Analyzing Nonblocking Switching Networks using Linear Programming (Duality)

Hung Q. Ngo, Atri Rudra, Thanh-Nhan Nguyen, Anh Le
Computer Science and Engineering
The State University of New York at Buffalo
Email: {hungngo, atri, nguyen9, anhle}@buffalo.edu

***Abstract*—The main task in analyzing a switching network design (including circuit-, multirate-, and photonic-switching) is to derive a formula for the minimum number of some switching components so that the design is non-blocking in some sense (e.g., strict-sense or wide-sense). We show that, in many cases, this task can be accomplished with a very simple and effective two-step strategy: (1) formulate a linear program whose optimum value is a bound for the minimum value we are seeking, and (2) specify a solution to the dual program, whose objective value by weak duality immediately yields a sufficient condition for the design to be non-blocking.**

We illustrate this technique through a variety of examples, ranging from circuit to multirate to photonic switching, from unicast to f-cast and multicast, and from strict-sense to wide-sense non-blocking. The switching architectures in the examples are of Clos-type and Banyan-type, which are the two most popular architectural choices for designing non-blocking switching networks.

To prove the result in the multirate Clos network case, we formulate a new problem called DYNAMIC WEIGHTED EDGE COLORING **which generalizes the** DYNAMIC BIN PACKING **problem. We then design an algorithm with competitive ratio 5.6355 for the problem. The algorithm is analyzed using the linear programming technique. We also show that no algorithm can have competitive ratio better than $4 - O(\log n/n)$ for this problem. New lower- and upper-bounds for multirate wide-sense non-blocking Clos networks follow.**

Keywords: Nonblocking, multirate, switching, linear programming, duality, dynamic weighted edge coloring.

I. Introduction

The two most important architectures for designing nonblocking switching networks are Clos-type [1] and Banyan-type [2]. The Clos networks not only played a central role in classical circuit-switching theory [3], [4], but also were the bedrock of multirate switching [5]–[10] (e.g., in time-divisioned switching environments where connections are of varying bandwidth requirements), and photonic-swtiching [11]–[15]. The Banyan network is equivalent to various other "bit-permutation" networks such as Omega, baseline, etc., they are called *Banyan-type* networks and have been used extensively in designing electronic and optical switches, as well as parallel processor architectures [16]. In particular, the multilog design which involves the vertical stacking of a number of inverse Banyan planes has been the architecture of choice in circuit- and photonic-switching environments because they have small depth ($\log N$), self-routing capability, and absolute signal loss uniformity [17]–[21].

In analyzing Clos networks, the most basic task is to determine the minimum number of middle-stage switches so that the network satisfies some nonblocking conditions. This holds true in space-, multirate-, and photonic-switching, in unicast, f-cast and multicast, and broadcast traffic patterns, and in all nonblocking types (strict-sense, wide-sense, and rearrangeable). Similarly, the basic task in analyzing multilog networks is to determine the minimum number of inverse Banyan planes so that the network satisfies some requirements.

This paper shows that a simple and effective linear programming (LP) based strategy can be used to accomplish the task. The strategy involves two steps:

- First, the minimum value we are seeking (e.g., the number of middle-stage switches in a Clos network or the minimum number of inverse Banyan planes in a multilog network) can be formulated so that it is upper-bounded by the optimum value of a linear program (LP) of the form $\max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. The maximization objective is often due to the fact that we are considering the worst case scenarios. For example, at the maximum how many middle-stage switches in a Clos network is insufficient to carry a new request. The constraints of the LP are used to express the fact that no input or output can generate or receive connections totaling more than its capacity.
- Second, by specifying any feasible solution, say $\mathbf{y}^*$, to the dual program $\min\{\mathbf{b}^T\mathbf{y} \mid \mathbf{A}^T\mathbf{y} \geq \mathbf{c}\}$, applying weak duality we can use the dual-objective value $\mathbf{b}^T\mathbf{y}^*$ as an upper bound for the minimum value being sought.

In a few cases, we may not need the dual program since the primal program is small with only a few variables. In most cases, however, the LP and its dual are very general, dependent on various parameters of the switch design (number of inputs and switching elements' dimensions, for example). Thus, we cannot hope to derive a formula for a primal-optimal solution and have to rely on a dual-feasible solution to quickly "certify" the bound. In fact, as we shall see later in this paper, we might need to specify a family of dual-feasible solutions and choose the best one (i.e. the one giving the smallest objective value) in accordance with the parameters of the problem at hand.

The LP-duality technique was first used in our recent paper [22] to analyze the (unicast) strictly nonblocking multilog architecture in the photonic-switching case, subject to general crosstalk constraits. This paper demonstrates that the technique can be applied to a much wider range of switching network design problems. Due to their pervasiveness, we use the Clos and multilog networks as canonical examples to illustrate our analysis technique.

Our main contributions are as follows. First, we formulate a new problem called DYNAMIC WEIGHTED EDGE COLORING (DWEC) of graphs, which generalizes the classic DYNAMIC BIN PACKING problem [23] and the routing problem for multirate widesense nonblocking Clos networks. Using the LP-technique, we design an algorithm with competitive ratio 5.6355. We also show that no algorithm can have competitive ratio better than $4 - O(\log n/n)$. New lower and upper bounds for the multirate Clos network problem follow trivially, improving upon a couple of 10-year old bounds. Since BIN PACKING and its variations are extremely useful in both theory and practice, we believe that DWEC and our results on it are of independent interest. Second, we use the LP-technique to prove a general theorem for the multilog network to be f-cast nonblocking with respect to the so-called *window algorithm*. To the best of our knowledge, this is the first f-cast result for the multilog design. Many known results are immediate consequences of our theorem.

The rest of this paper is organized as follows. Section II fixes notations and terminologies. Section III illustrates the strength of the LP-duality technique on analyzing several problems on the Clos networks. The DWEC problem is also defined and analyzed. Section IV addresses the multilog architectures. Section V concludes the paper with a few remarks.

II. Preliminaries

Throughout this paper, for any positive integers k, d, let $[k]$ denote the set $\{1, \dots, k\}$, $\mathbb{Z}_d$ denote the set $\{0, \dots, d-1\}$ which can be thought of as d-ary "symbols", $\mathbb{Z}_d^k$ denote the set of all d-ary strings of length k, $|\mathbf{s}|$ the length of any d-ary string $\mathbf{s}$ (e.g., $|31| = 2$), $\mathbf{s}_{i..j}$ the substring $s_i \cdots s_j$ of a string $\mathbf{s} = s_1 \dots s_l \in \mathbb{Z}_d^l$ (if $i > j$ then $\mathbf{s}_{i..j}$ is the empty string).

There are three nonblockingness-degrees of a switching network: rearrangeably nonblocking (RNB), widesense nonblocking (WSNB), and strictly nonblocking (SNB). The reader is referred to [4] for their precise definitions.

A. Switching environments

Consider an $N \times N$ switching network, i.e. a switching network with N inputs and N outputs. In circuit switching, a request is a pair $(\mathbf{a}, \mathbf{b})$ where $\mathbf{a}$ is an unused input and $\mathbf{b}$ is an unused output. A route $R(\mathbf{a}, \mathbf{b})$ realizes the request if it does not share any internal link with existing routes.

In the multirate case, each link has a capacity (e.g., bandwidth). All inputs and outputs have the same capacity normalized to 1. An input cannot request more than its capacity, so do outputs. A request is of the form $(\mathbf{a}, \mathbf{b}, w)$ where $\mathbf{a}$ is an input, $\mathbf{b}$ is an output, and $w \leq 1$ is the requested *rate*. If existing requests have used up to x and y units of $\mathbf{a}$'s and $\mathbf{b}$'s capacity, respectively, then the new requested rate w can only be at most $\min\{1-x, 1-y\}$. An internal link cannot carry requests with total rate more than 1.

B. The 3-stage Clos networks

The Clos network $C(n_1, r_1, m, n_2, r_2)$ is a 3-stage interconnection network, where the first stage consists of r_1 crossbars of size $n_1 \times m$, the last stage has r_2 crossbars of dimension $m \times n_2$, and the middle stage has m crossbars of dimension $r_1 \times r_2$ (see Figure 1). Each input crossbar I_i $(i = 1, \dots, r_1)$ is connected to each middle crossbar M_j $(j = 1, \dots, m)$. Similarly, the middle stage and the last stage are fully connected. When $n_1 = n_2 = n$ and $r_1 = r_2 = r$, the network is called the *symmetric* 3*-stage Clos network*, denoted by $C(n, m, r)$.

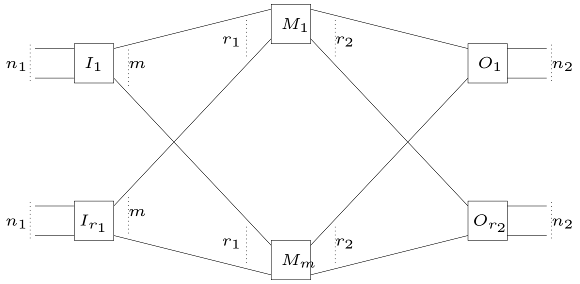

Fig. 1. The 3-stage Clos network $C(n_1, r_1, m, n_2, r_2)$

C. The d-ary multilog networks

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 $\text{BY}^{-1}(n)$ with N inputs and N outputs. (See Fig. 2 and 3.) Label the inputs and outputs of $\text{BY}^{-1}(n)$ and the $d \times d$ switching elements (SE) of each stage of $\text{BY}^{-1}(n)$ as illustrated in Fig. 2. Specifically, each input $\mathbf{u} \in \mathbb{Z}_d^n$ and output $\mathbf{v} \in \mathbb{Z}_d^n$ have the form $\mathbf{u} = u_1 \cdots u_n$, $\mathbf{v} = v_1 \cdots v_n$, where $u_i, v_i \in \mathbb{Z}_d$, $\forall i \in [n]$. Similarly, the $d \times d$ SEs in each of the n stages of $\text{BY}^{-1}(n)$ are labeled with d-ary strings of length $n-1$.

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. The following proposition is immediate (for more details, see [24], e.g.).

Proposition II.1. *Let $(\mathbf{a}, \mathbf{b})$ and $(\mathbf{u}, \mathbf{v})$ be two unicast requests. Then their corresponding routes $R(\mathbf{a}, \mathbf{b})$ and $R(\mathbf{u}, \mathbf{v})$ in a BY$^{-1}(n)$-plane share at least a common link if and only if*

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

III. Results on the Clos Networks

A. A classic example in circuit switching

This example is a classic result by Benes [3]. Consider the $C(n, m, 2)$ network. The routing algorithm is to reuse a busy middle switch if possible. For any $i, j \in \{1, 2\}$, let M_{ij} be the set of middle switches carrying an I_i, O_j-request. The sets M_{ij} certainly change over time as requests come and go. However, it is easy to show by induction that the routing rule ensures

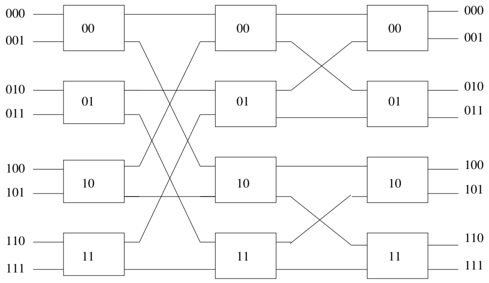

Fig. 2. The inverse Banyan network $\mathrm{BY}^{-1}(3)$

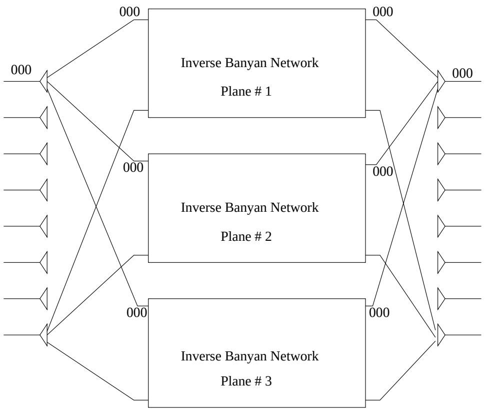

Fig. 3. A multi-log network with 3 inverse Banyan planes

$|M_{11} \cup M_{22}| \le n$ and $|M_{12} \cup M_{21}| \le n$ at all times. Without loss of generality, consider a new request from I_1 to O_1. If $M_{22} \setminus M_{11} \neq \emptyset$, then we have a busy switch to reuse. Otherwise, the number of unavailable middle-switches for this new request is precisely $|M_{11} \cup M_{12} \cup M_{21}| = |M_{11}| + |M_{12} \cup M_{21}|$. Just before the arrival of this new request, the number of existing requests from I_1 or to O_1 is at most $n-1$, i.e. $|M_{11} \cup M_{12}| = |M_{11}| + |M_{12}| \le n-1$, and $|M_{11} \cup M_{21}| = |M_{11}| + |M_{21}| \le n-1$. The number of unavailable middle switches is thus bounded by the optimal value of the following LP (think of set cardinalities as variables):

$$\begin{array}{lcl}
\max & |M_{11}| + |M_{12} \cup M_{21}| & \\
\text{s.t.} & |M_{11}| + |M_{12}| & \le n-1 \\
 & |M_{11}| + |M_{21}| & \le n-1 \\
 & |M_{12}| + |M_{21}| & \le n \\
 & |M_{12} \cup M_{21}| - |M_{21}| - |M_{21}| & \le 0
\end{array}$$

The last inequality is the straightforward union bound. Obviously, all cardinalities are non-negative. The dual LP is

$$\begin{array}{lcl}
\min & (n-1)(y_1+y_2) + ny_3 & \\
\text{s.t.} & y_1 + y_2 & \ge 1 \\
 & y_2 + y_3 - y_4 & \ge 0 \\
 & y_1 + y_3 - y_4 & \ge 0 \\
 & y_4 \ge 1, \quad y_1, y_2, y_3 & \ge 0
\end{array}$$

Setting $y_1 = y_2 = y_3 = 1/2$ and $y_4 = 1$ is certainly feasible with objective value $3n/2 - 1$. Hence, the number of unavailable middle-switches for a new request is at most $\lfloor 3n/2 \rfloor - 1$, which means $m \ge \lfloor 3n/2 \rfloor$ is sufficient for $C(n,m,2)$ to be WSNB. It's known that $m \ge \lfloor 3n/2 \rfloor$ is necessary, too.

B. Multirate switching and the DWEC *problem*

It has been known for more than 12 years that $C(n,m,r)$ is multirate WSNB if $m \ge 5.75n$ [10]. This section uses the LP technique to improve this bound via solving a much more general problem called DYNAMIC WEIGHTED EDGE COLORING DWEC.

Definition III.1 (The DWEC problem)**.** Let $G = (V,E)$ be a fixed simple graph called the *base graph*. Let $G_0 = (V, \emptyset)$ be an empty graph with the same vertex set. At time t, either an arbitrary edge is removed from G_{t-1} or a copy of some edge $e \in E$ "arrives" along with a weight $w_e \in (0,1]$. Define $G_t = G_{t-1} \cup \{e\}$ and note that G_t in general is a multi-graph as many copies of the same edge may arrive over time. The arriving edge is to be colored so that, in G_t, the total weight of same-color edges incident to the same vertex is at most 1.

The objective is to design a coloring algorithm so that the number of colors used is minimized, compared to an off-line algorithm which colors edges of G_t subject to the same constraint. Formally, let $\mathrm{OPT}(t)$ denote the number of colors used by an optimal off-line algorithm on G_t. Let $\overline{\mathrm{OPT}}(t) = \max_{i \le t} \mathrm{OPT}(i)$. For any online coloring algorithm $\mathbf{A}$, let $\bar{\mathbf{A}}(t)$ be the number of colors ever used by $\mathbf{A}$ up to time t. Algorithm $\mathbf{A}$ has *competitive ratio* ρ if, for any sequence of edge arrivals/departures with arbitrary weights, we always have $\bar{\mathbf{A}}(t) \le \rho \cdot \overline{\mathrm{OPT}}(t), \forall t$.

The DYNAMIC BIN PACKING problem is *exactly* the DWEC problem when the base graph $G = K_2$, where each color is a bin. The best competitive ratio for DYNAMIC BINPACKING is known to be between 2.5 and 2.788 [23]. We will show that the DWEC's best competitive ratio is somewhere between 4 and 5.6355.

Theorem III.2. *There is an algorithm for* DWEC *with competitive ratio* 5.3655*.*

Proof: For the sake of presentation clarity, we will prove a slightly weaker ratio of 5.675, and then indicate how to obtain the better ratio 5.6355. The two proofs are identical, but the one we present is cleaner.

At any time t, let $W^u(t)$ denote the total weight of edges incident to u in G_t, and let $d^u(t)$ denote the number of edges of weight $> 1/2$ incident to u. Let $\overline{W}(t) = \max_t \max_u W^u(t)$ and $\overline{\Delta}(t) = \max_t \max_u d^u(t)$. Then, it is easy to see that $\lceil \overline{W}(t) \rceil \le \overline{\mathrm{OPT}}(t)$ and $\overline{\Delta}(t) \le \overline{\mathrm{OPT}}(t)$.

Refer to an edge a *type*-0, *type*-1, *type*-2, or *type*-3, if its weight belongs to the interval $(\frac{1}{2}, 1]$, $(\frac{2}{5}, \frac{1}{2}]$, $(\frac{1}{3}, \frac{2}{5}]$, $(0, \frac{1}{3}]$, respectively. Our coloring algorithm is as follows. Maintain 4 *disjoint* sets of colors $C_i(t)$, $0 \le i \le 3$. Let x_0, x_1, x_2, x_3 be constants to be determined. For each $i = 0..3$, we will maintain the following time-invariant conditions: $|C_i(t)| = \lceil x_i \overline{W}(t) \rceil$ for $1 \le i \le 3$ and $|C_0(t)| = \lceil x_0 \overline{\Delta}(t) \rceil$.

If $\overline{W}(t)$ or $\overline{\Delta}(t)$ is increased at some time t then we are allowed to add colors to the $C_i(t)$ to to maintain the above invariant conditions. Note that $\overline{W}(t)$ is a non-decreasing function in t; hence, colors will never be removed from the $C_i(t)$. The colors in $C_0(t)$ are used exclusively for edges of type-0. The coloring for edges of types i, $1 \le i \le 3$ is done as follows. If a type-i edge arrives at time t, find a color in $C_i(t)$ to color it. If $C_i(t)$ cannot accommodate this edge, try $C_{i+1}(t)$, and so on until $C_3(t)$.

We next show that if the constants x_i are feasible solutions to some certain LP, then it is always possible to find a color for an arriving edge.

Suppose a type-0 edge $e = (u, v)$ arrives at time t. If we cannot find a color in $C_0(t)$ for e, then $|C_0(t)| \le d^u(t-1) + d^v(t-1) = (d^u(t) - 1) + (d^v(t) - 1) < 2\overline{\Delta}(t)$. Hence, as long as $x_0 \ge 2$ we can color e.

Next, suppose $e = (u, v)$ of type 1 is arriving at time t and we cannot find a color in $C_1(t) \cup C_2(t) \cup C_3(t)$ to color e. For a color $c \in C_1(t)$ to be unavailable for e, there must be at least two type-1 color-c edges incident to either u or v. Thus, the total type-1 weight at u and v is at least $\frac{4}{5}|C_1(t)| = \frac{4}{5}\lceil x_1\overline{W}(t)\rceil$. Similarly, for each color c in $C_2(t)$, the total c-weight incident to u and v must be $> 1/2$, which means this color c "carries" either at least two type-1 edges, or one type-1 edge and one type 2 edge, or at least two type-2 edges. Thus, the total color-c weight incident to u and v must be at least $\frac{2}{3}\lceil x_2\overline{W}(t)\rceil$. Lastly, for each color c in $C_3(t)$, the total color-c weight incident to u and v must be $> 1/2\lceil x_3\overline{W}(t)\rceil$. Note that the total weight at u and v is $< 2\overline{W}(t)$. Consequently, we will be able to find a color for e if

$$\frac{4}{5}|C_1(t)| + \frac{2}{3}|C_2(t)| + \frac{1}{2}|C_3(t)| \ge 2\overline{W}(t),$$

which would hold if $\frac{4}{5}x_1 + \frac{2}{3}x_2 + \frac{1}{2}x_3 \ge 2$. Similarly, a newly arriving type-2 edge is colorable if $\frac{2}{3}x_2 + \frac{3}{5}x_3 \ge 2$, and a new type-3 edge is colorable if $\frac{2}{3}x_3 \ge 2$.

Consequently, our coloring algorithm works if the x_i are feasible for the following LP:

$$\begin{array}{llllllll}
\min & x_0 & +x_1 & +x_2 & +x_3 & & \\
\text{s.t.} & x_0 & & & & \ge & 2 \\
& & \frac{4}{5}x_1 & +\frac{2}{3}x_2 & +\frac{1}{2}x_3 & \ge & 2 \\
& & & \frac{2}{3}x_2 & +\frac{3}{5}x_3 & \ge & 2 \\
& & & & \frac{2}{3}x_3 & \ge & 2 \\
& & & & x_0, x_1, x_2, x_3 & \ge & 0.
\end{array}$$

The solution $x_0 = 2, x_1 = 3/8, x_2 = 3/10, x_3 = 3$ is certainly feasible. The total number of colors used is

$$\begin{aligned}
&\lceil x_0\overline{\Delta}(t)\rceil + \sum_{i=1}^{3}\lceil x_i\overline{W}(t)\rceil \\
&\quad \le (x_0 + x_1 + x_2 + x_3)\overline{\text{OPT}}(t) + 4 = 5.675\overline{\text{OPT}}(t) + 4.
\end{aligned}$$

As is customary in online/dynamic algorithm analysis, we ignore the constant term of 4, as we let $\overline{\text{OPT}}(t) \to \infty$.

To prove the better ratio 5.6355, we divide the rates into 5 types belonging to the intervals $(1/2, 1], (2/5, 1/2], (1/3, 2/5], (11/43, 1/3]$, and $(0, 11/43]$. The proof is pretty much identical. ■

Corollary III.3. *The Clos network $C(n, m, r)$ is multirate WSNB if $m \ge 5.6355n$.*

Proof: Consider the multirate WSNB problem on the Clos network $C(n, m, r)$. The base graph is the complete bipartite graph $G = \mathcal{I} \times \mathcal{O}$, where $\mathcal{I}$ is the set of input switches and $\mathcal{B}$ is the set of output switches. When a new request $(\mathbf{a}, \mathbf{b})$ with rate w arrives at time t, add an edge $e = (I, O)$ to G_{t-1} where I is the input switch that $\mathbf{a}$ belongs and O is the output switch that $\mathbf{b}$ belongs. The weight w_e of the new edge is simply w. Think of each middle-stage switch as a color. Obviously, the maximum number of colors ever used by an algorithm $\mathbf{A}$ is also the number of middle switches needed for $C(n, m, r)$ to be non-blocking.

In the above algorithm, $\overline{\Delta}(t) \le n$ because the number of requests with rate $> 1/2$ coming out of the same input switch or into the same output switch is at most n (one per input/output). Moreover, $\overline{W}(t) \le n$ because the total rate of requests from/to and input/output is at most n. Hence, the number of middle-stage switches (i.e. colors) needed is at most $5.6355n + 4$. ■

Remark III.4. Our strategy can also give a better sufficient condition than the best known in [10] for the case when there's internal speed up in the Clos network. However, the keep the exposition clean and thus easier to follow, we refrain from stating the most general result we can prove.

Theorem III.5. *For every large enough $n \ge 1$, there exists a base graph G such that* every *algorithm for* DWEC *has competitive ratio at least $4 - O(\log n/n)$. This lower bound holds even if there are only two distinct edge weights.*

Proof: We present an adversarial strategy that forces **any** algorithm to use at least $4n - 2\lceil\log n\rceil - 8$ colors while the optimal solution uses at most n colors. For the ease of exposition, we will not specify the base graph G upfront but it can be deduced from the adversarial strategy below. Further, $G = (V, E)$ will be a bipartite graph. For the rest of the proof fix an arbitrary algorithm $\mathbf{A}$. For simplicity, we will assume that n is a power of two.

The proof has two phases. In the first phase there are only edges of weight $1/n$ (*light* edges), and if necessary in the second phase we introduce edges of weight 1 (*heavy* edges). Let $\Delta_l(t)$ be the maximum light degree at time t and $\Delta_h(t)$ be the maximum heavy degree at time t. Since the base graph is bipartite, by König theorem the heavy edges can be colored with $\Delta_h(t)$ colors. The light edges can be colored with $\Delta_l(t)$ colors so that edges incident to the same vertex are colored differently. However, since light edges have a weight of $1/n$, we can combine n light colors into one. Hence, $\text{OPT}(t) \le \Delta_h(t) + \lceil\Delta_l(t)/n\rceil$. This fact is implicit throughout the proof.

First, we claim that there exists a sequence of light edge arrivals/departures and a time t such that $\overline{\text{OPT}}(t) \le n$ and either (i) $\overline{\mathbf{A}}(t) \ge 4n$ or (ii) G_t contains the following subgraph:

there exists m pairs of distinct vertices $(u_1,v_1),\ldots,(u_m,v_m)$ (m to be determined) such that for every $1\le i\le m$, every edge incident to u_i (v_i resp.) is assigned a unique color by $\mathbf{A}$ from the same set S^u (S^v resp.) where $S^v\cap S^u=\emptyset$ and $|S^u|=|S^v|=n-\log n-1$. Assuming this claim, we finish the rest of the proof.

If (i) is true then we are done. So now assume that (ii) holds and we move onto the second phase which has a sequence of heavy edge arrivals and departures such that at some time $t'>t$, $\overline{\text{OPT}}(t')\le n$ but $\overline{\mathbf{A}}(t')\ge 4n-2\log n-8$. For every $1\le i\le m$, pick a "fresh" vertex $w_i\in V$ such that none of the previous edges were incident on w_i and add $n/2-3$ copies of the edge (u_i,w_i) and $n/2-3$ copies of the edge (v_i,w_i). Let t'' be the time when the last such edge arrives for $i=m$. Note that by our choice of edge arrivals, we will maintain the invariant that $\overline{\text{OPT}}(t'')\le n$. Further, each of these new edges cannot use any color from $S^u\cup S^v$.

Now if $\overline{\mathbf{A}}(t'')\ge 4n$, then we are done. Otherwise, we claim that there exists at least $n/2$ distinct pairs $(u_{i_1},v_{i_1}),\ldots,(u_{i_{n/2}},v_{i_{n/2}})$ ($i_j\in[m],\forall j\in[n/2]$) such that the edges incident on any pair (u_{i_j},v_{i_j}) contain the same set T of $3n-2\log n-8$ colors. Let's finish the proof before proving the claim. Let $w^*\in V$ be such that no edge seen till now has been incident on it. Now the adversary adds one copy each of the edges (u_{i_j},w^*) and (v_{i_j},w^*) for every $1\le j\le n/2$. Let t' be the time when the last such edge is added for $j=n/2$. Note that by construction, $\overline{\text{OPT}}(t')\le n$ and n new edges are added. Further, each of these n edges have to be assigned a brand new color. Thus, we have $\overline{\mathbf{A}}(t')\ge 4n-2\log n-8$.

Next, we argue the existence of the pairs $(u_{i_1},v_{i_1}),\ldots(u_{i_{n/2}},v_{i_{n/2}})$. Recall $\overline{\mathbf{A}}(t'')<4n$. Further, every pair (u_i,v_i) ($1\le i\le m$) is assigned a set of color T_i of size exactly $3n-2\log n-8$. Now the number of possible distinct T_i's is upper bounded by $\binom{4n}{3n-2\log n-8}<2^{6n}$ (for large enough n). Thus, the existence of the pairs $(u_{i_1},v_{i_1}),\ldots,(u_{i_{n/2}},v_{i_{n/2}})$ will follow by the pigeon-hole principle if we choose $m=n\cdot 2^{6n-1}$.

To complete the proof, we now show the existence of the pairs $(u_1,v_1),\ldots,(u_m,v_m)$ with the required property. Towards this end we give an inductive proof to show that for every $1\le\ell\le\log n$, we have m_ℓ pairs of vertices $(u_1^{(\ell)},v_1^{(\ell)}),\ldots,(u_{m_\ell}^{(\ell)},v_{m_\ell}^{(\ell)})$ at the end of time t_ℓ such that

(i) Either $A(t_\ell)$ uses at least $4n$ colors; or

(ii) There exists disjoint sets C_ℓ^u and C_ℓ^v of colors with the following properties for every $1\le j\le m_\ell$:

(a) Every edge incident on $u_j^{(\ell)}$ ($v_j^{(\ell)}$ resp.) uses a color from C_ℓ^u (C_ℓ^v resp.).

(b) Every color $c\in C_\ell^u\cup C_\ell^v$ is used n_c^ℓ times for the pair $(u_j^{(\ell)},v_j^{(\ell)})$. That is, every $u_{(\cdot)}^{(\ell)}$ (and every $v_{(\cdot)}^{(\ell)}$) has the same "spectrum."

(c) $C_{\ell-1}^u\subset C_\ell^u$ and $C_{\ell-1}^v\subset C_\ell^v$. Further, for $w\in\{u,v\}$:

$$\sum_{k=1}^{\ell}\frac{n}{2^k}-\ell\le\frac{1}{n}\cdot\sum_{c\in C_\ell^w}n_c^\ell\le\sum_{k=1}^{\ell}\frac{n}{2^k}. \tag{2}$$

Finally, the total weight of edges at $u_j^{(\ell)}$ and $v_j^{(\ell)}$ are at most n.

As we will see later $m_\ell<m_{\ell+1}$ and we will set $m_{\log n}=m$. If for any $1\le\ell\le\log n$, (i) is satisfied then we set $t=t_\ell$ and stop. Otherwise, we output the remaining m pairs $(u_1^{(\log n)},v_1^{(\log n)}),\ldots(u_m^{(\log n)},v_m^{(\log n)})$. By property (ii)(c), we have $|C_{\log n}^u|,|C_{\log n}^v|=n-\log n-1$ (the sizes could be larger in which case we drop the "extra" colors). For every color $c\in S_\ell^u\cup S_\ell^v$ retain exactly one edge of color c. Thus, the m pairs have the required property.

We are done except for demonstrating a sequence of light edge arrivals and departures such that the adversary can maintain the required m_ℓ pairs. For the rest of the proof, we will assume that property (i) is never satisfied (as otherwise we are done).

We begin with the base case of $\ell=1$. For a parameter N (to be determined soon), let $\{x_1,\ldots,x_{2N}\}\subset V$ and $\{y_1,\ldots,y_{2N}\}\subset V$ be disjoint subsets of vertices. The adversary adds n^2 copies of the edges (x_k,y_k) (for every $1\le k\le 2N$), each with weight $1/n$. Now if we pick $N=n^{4n}\cdot m_1$, then by the pigeonhole principle there exists $2m_1$ pairs $(x_{i_1},y_{i_1}),\ldots,(x_{i_{2m_1}},y_{i_{2m_1}})$ such that they have the same "spectrum" (i.e. all the pairs have the same color set C_1 and each color is used the same number of times). It is easy to see that C_1 can be divided into two disjoint subsets C_1^u and C_1^v such that every pair (x_{i_j},y_{i_j}) (for $1\le j\le 2m_1$), the number of edges colored with a color in C_1^u or C_1^v is in the range $(n^2/2-n,n^2/2]$. For every odd (even resp.) j, throw away edges (x_{i_j},y_{i_j}) that are colored with a color from C_1^v (C_1^u resp.) and define $u_{(j+1)/2}^{(1)}=x_{i_j}$ ($v_{j/2}^{(1)}=y_{i_j}$ resp.). It can be verified that the pairs constructed above satisfy the properties (ii)(a)-(ii)(c).

The argument for the inductive step is similar to the $\ell=1$ case. Define $m_\ell=2n^{4n}m_{\ell+1}$. Given m_ℓ pairs $(u_1^{(\ell)},v_1^{(\ell)}),\ldots,(u_{m_\ell}^{(\ell)},v_{m_\ell}^{(\ell)})$, add $n^2/2^\ell$ edges between the u vertices of the odd pair and the v vertices of the (immediate next) even pair. Using similar argument as with the $\ell=1$ on these new edges, we can argue the existence of the required pairs $(u_j^{(\ell+1)},v_j^{(\ell+1)})$ for $1\le j\le m_{\ell+1}$. ■

A similar proof gives us the following improved bound. The previous best known bound was $3n-3$ [25].

Corollary III.6. *The Clos network $C(n,m,r)$ is not multirate WSNB if $m<4n-O(\log n/n)$ and r is sufficiently large.*

IV. Results on the d-ary Multilog Networks

Let f,t be a given integers with $0\le t\le n$, and $1\le f\le N=d^n$. In this subsection we analyze f-cast WSNB $\log_d(N,0,m)$ networks under the *window algorithm* with window size d^t. The algorithm was proposed and analyzed for one window size $d^{\lfloor n/2\rfloor}$ in [26], and later analyzed more carefully for varying window sizes in [27]. Both papers considered the broadcast case. We will derive a more general theorem for the f-cast case.

- **The Window Algorithm with window size d^t:** Given any integer t, $0\le t\le n$, divide the outputs into "windows" of size d^t each. Each window consists of all outputs sharing a prefix of length $n-t$, for a total of d^{n-t} windows. Denote the windows as $W_w, 0 \leq w \leq d^{n-t}-1$. Given a new multicast request $(\mathbf{a}, B)$, where $\mathbf{a}$ is an input and B is a subset of outputs, the routing rule is, for every $0 \leq w \leq d^{n-t}-1$, the subrequest $(\mathbf{a}, B \cap W_w)$ is routed entirely on one single $\mathrm{BY}^{-1}(n)$ plane. (Different sub-requests can be routed through the same or different $\mathrm{BY}^{-1}(n)$-planes.)

Remark IV.1. there is a subtle point about the window algorithm due to which the original authors in [26] thought their multilog network was SNB instead of WSNB. Basically, for some specific values of the parameters the algorithm is **no** algorithm at all. In those cases, any sufficient condition for the network to be nonblocking under the window algorithm is in fact an SNB condition, not a WSNB condition. Obvious examples include the unicast ($f = 1$) case and the $t = 0$ case. Yet another example is when $t = n$. Here, the routing rule is for each request to be routed entirely on some plane. If the $1 \times m$-SE stage of the multilog network has fanout capability, then the rule **does** restrict how we route requests, and thus we indeed have a WSNB situation. However, if the $1 \times m$-SE stage is implemented with $1 \times m$-demultiplexors, then we *have to* route each request entirely on some plane. Thus, any sufficient condition is an SNB condition.

A. Setting up the linear program and its dual

Let $(\mathbf{a}, B)$ be an arbitrary f-cast request to be routed using the window algorithm with window size d^t. Following the window algorithm, due to symmetry without loss of generality we can assume that $B = \{\mathbf{b}^{(1)}, \dots, \mathbf{b}^{(k)}\}$ where all the outputs $\mathbf{b}^{(l)}$ ($l \in [k]$) belong to the same window W_0, and $k \leq \min\{f, d^t\}$. The $\mathbf{b}^{(l)}$ thus share a common prefix of length $n-t$.

Define

$$A_i := \{\mathbf{u} \in \mathbb{Z}_d^n - \{\mathbf{a}\} \mid \mathrm{SUF}(\mathbf{u}_{1..n-1}, \mathbf{a}_{1..n-1}) = i\},$$

and

$$B_j := \left\{\mathbf{v} \in \mathbb{Z}_d^n - B \mid \exists l \in [k], \mathrm{PRE}(\mathbf{v}_{1..n-1}, \mathbf{b}^{(l)}_{1..n-1}) = j\right\}.$$

Note that $|A_i| = d^{n-i} - d^{n-1-i}$, for all $0 \leq i \leq n-1$, and $|B_j| = d^{n-j} - d^{n-1-j}$ for all $0 \leq j \leq n-t-1$. Note that $\bigcup_{j=n-t}^{n-1} B_j \subseteq W_0$. Define $\mathcal{A} = \bigcup_{i=0}^{n-1} A_i$, $\mathcal{B} = \bigcup_{j=0}^{n-t-1} B_j$. For every input $\mathbf{u} \in \mathcal{A}$, let $i(\mathbf{u})$ denote the index i such that $\mathbf{u} \in A_i$.

For each $j \leq n-t-1$, B_j is the disjoint union of precisely $d^{n-j-t} - d^{n-1-j-t}$ windows of size d^t each. In fact, it is easy to see that $\bigcup_{j=0}^{n-t-1} B_j = \bigcup_{w=1}^{d^{n-t}-1} W_w$, and $\bigcup_{j=n-t}^{n-1} B_j \subseteq W_0$. For every $w \in [d^{n-t}]$, let $j(w)$ be the index j such that $W_w \subseteq B_j$.

Lemma IV.2. *For each input $\mathbf{u} \in \mathcal{A}$ and each $w \in \{0, \dots, d^{n-t}-1\}$ such that $i(\mathbf{u}) + j(w) \geq n$, define a variable $x_{\mathbf{u},w}$. Also, for each input $\mathbf{u} \in \mathcal{A}$ and each output $v \in W_0 - B$ where there exists a $j \geq n-t$ such that $\mathbf{v} \in B_j$ and $i(\mathbf{u}) + j \geq n$, define a variable $x_{\mathbf{u},\mathbf{v}}$. Then, the number of planes blocking $(\mathbf{a}, B)$ is upperbounded by the optimal value of the following LP:*

$$\begin{array}{llcll}
\max & \displaystyle\sum_{\mathbf{u},w} x_{\mathbf{u},w} + \sum_{\mathbf{u},\mathbf{v}} x_{\mathbf{u},\mathbf{v}} & & & \\
\text{s.t.} & \sum_{\mathbf{u}} x_{\mathbf{u},w} & \leq & d^t & w \in \{0, \dots, d^{n-t}-1\} \\
 & x_{\mathbf{u},w} & \leq & 1 & \forall \mathbf{u}, w \\
 & \sum_{\mathbf{v}} x_{\mathbf{u},\mathbf{v}} & \leq & 1 & \forall \mathbf{u} \in \mathcal{A} \\
 & \sum_{\mathbf{u}} x_{\mathbf{u},\mathbf{v}} & \leq & 1 & \forall \mathbf{v} \in W_0 - B \\
 & \sum_w x_{\mathbf{u},w} + \sum_{\mathbf{v}} x_{\mathbf{u},\mathbf{v}} & \leq & f & \forall \mathbf{u} \in \mathcal{A} \\
 & x_{\mathbf{u},w}, x_{\mathbf{u},\mathbf{v}} & \geq & \mathbf{0} & \forall \mathbf{u}, w, \mathbf{v}
\end{array} \tag{3}$$

Obviously, the sums and the constraints only range over values for which the variables are defined.

Proof: Suppose the network $\log_d(N, 0, m)$ already had some routes established. Consider a $\mathrm{BY}^{-1}(n)$-plane which blocks the new request $(\mathbf{a}, B)$. There must be one route $R(\mathbf{u}, \mathbf{v})$ on this plane for which $R(\mathbf{u}, \mathbf{v})$ and $R(\mathbf{a}, \mathbf{b}^{(l)})$ share a link, for some $l \in [k]$. Note that the branch $R(\mathbf{u}, \mathbf{v})$ could be part of a multicast tree from input $\mathbf{u}$, but we only need an arbitrary blocking branch $(\mathbf{u}, \mathbf{v})$ of this tree. Assemble one blocking branch $(\mathbf{u}, \mathbf{v})$ per blocking plane into a set S. Then, the number of blocking planes is $|S|$.

Fact 1: if $(\mathbf{u}, \mathbf{v})$ and $(\mathbf{u}, \mathbf{v}')$ are both in S then $\mathbf{v}$ and $\mathbf{v}'$ must belong to different windows; because, if they belong to the same window, the window algorithm would have routed them through the same plane, and S only contains one branch per blocking plane.

Fact 2: each output $\mathbf{v}$ can only appear once in S, because each output can only be part of at most one existing request.

Fact 3: if $(\mathbf{u}, \mathbf{v}) \in S$, then $(\mathbf{u}, \mathbf{v}) \in A_i \times B_j$ for some $i + j \geq n$, thanks to Proposition II.1.

Straightforwardly, we will show that S defines a feasible solution to the LP with objective value precisely $|S|$. Set $x_{\mathbf{u},w} = 1$ if there is some $(\mathbf{u}, \mathbf{v}) \in S$ such that $\mathbf{v} \in W_w$; and $x_{\mathbf{u},\mathbf{v}} = 1$ if there is some $(\mathbf{u}, \mathbf{v}) \in S$ such that $\mathbf{v} \in W_0 - B$. All other variables are set to 0. Due to Fact 3, the procedure does not set value for an undefined variable. Certainly $|S|$ is equal to the objective value of this solution.

We next verify that the solution satisfies all the constraints. The first constraint expresses the fact that each output in a window W_w of size d^t only appears at most once in S (Fact 2). The second and third constraints are a restatement of Fact 1. The forth constraint says that each output $\mathbf{v} \in W_0 - B$ appear at most once in S (Fact 2 again). The fifth constraint says that each input can only be part of at most f members of S, due to the f-cast nature of the network. ■

The dual LP can be written as follows.

$$\begin{array}{llcll}
\min & \displaystyle\sum_w d^t \alpha_w + \sum_{\mathbf{u},w} \beta_{\mathbf{u},w} + \sum_{\mathbf{u}} \gamma_{\mathbf{u}} + \sum_{\mathbf{v}} \delta_{\mathbf{v}} + \sum_{\mathbf{u}} f \epsilon_{\mathbf{u}} & & & \\
\text{s.t.} & \alpha_w + \beta_{\mathbf{u},w} + \epsilon_{\mathbf{u}} & \geq & 1, & x_{\mathbf{u},w} \text{ defined (DC-1)} \\
 & \gamma_{\mathbf{u}} + \delta_{\mathbf{v}} + \epsilon_{\mathbf{u}} & \geq & 1, & x_{\mathbf{u},\mathbf{v}} \text{ defined (DC-2)} \\
 & \alpha_w, \beta_{\mathbf{u},w}, \gamma_{\mathbf{u}}, \delta_{\mathbf{v}}, \epsilon_{\mathbf{u}} & \geq & 0 & \forall \mathbf{u}, \mathbf{v}, w
\end{array} \tag{4}$$

Note that the dual-constraints only exist over all $\mathbf{u}, \mathbf{v}, w$ for which $x_{\mathbf{u},w}$ and $x_{\mathbf{u},\mathbf{v}}$ are defined.

B. Specifying a family of dual-feasible solutions

To illustrate the technique, let us first derive a known result "for free."

Corollary IV.3 (Theorem III.3 in [24]). *Suppose the demux-stage of the* $\log_d(N,0,m)$ *network stage does not have fanout capability, then it is SNB iff* $m \geq f\left(d^{\left\lceil \frac{n-r-2}{2} \right\rceil} - 1\right) + d^{n-\left\lceil \frac{n-r}{2} \right\rceil}$, *where* $r = \lfloor \log_d f \rfloor$.

Proof: Routing using the window algorithm with window size $t = n$ is the same as routing arbitrarily in the network when the demux stage does not have fanout capability. That is why any sufficient condition for the window algorithm to work is an SNB (rather than a WSNB) condition. Consider a solution to the dual LP as follows. Define $q = \left\lceil \frac{n-r}{2} \right\rceil$. Set $\gamma_{\mathbf{u}} = 1$ for all $\mathbf{u}$ with $i(\mathbf{u}) \geq q$ and $\delta_{\mathbf{v}} = 1$ for all $\mathbf{v} \in \bigcup_{j=n-q+1}^{n-1} B_j$. All other dual variables are 0. The solution is dual feasible with objective value $f\left(d^{\left\lceil \frac{n-r-2}{2} \right\rceil} - 1\right) + d^{n-\left\lceil \frac{n-r}{2} \right\rceil} - 1$. Hence, one more plane is sufficient to route the new (arbitrary) request. ■

The above corollary solves the $t = n$ case. We will consider $0 \leq t < n$ henceforth. We next specify a family of dual-feasible solution to the dual-LP (4). The main remaining task will be simple calculus as we pick the best dual-feasible solution depending on the parameters f, n, d, t of the problem.

The family of dual-feasible solution is specified with two integral parameters where $0 \leq p \leq n-t-1$ and $n-t \leq q \leq n$. The parameter p is used to set the variables $\epsilon_{\mathbf{u}}$, α_w and $\beta_{\mathbf{u},w}$, and the parameter q is used to set the variables $\gamma_{\mathbf{u}}$ and $\delta_{\mathbf{v}}$. As we set the variables, we will also verify the feasibility of the constraints (DC-1) and (DC-2), and the contributions of those variables to the final objective value.

Specifying the $\epsilon_{\mathbf{u}}$ variables. Set $\epsilon_{\mathbf{u}} = 1$ if $i(\mathbf{u}) \geq n-p$ and 0 otherwise. The contribution of the $\epsilon_{\mathbf{u}}$ to the dual objective is

$$\sum_{\mathbf{u}} f\epsilon_{\mathbf{u}} = \sum_{i=n-p}^{n-1} f \sum_{\mathbf{u}: i(\mathbf{u})=i} 1 = \sum_{i=n-p}^{n-1} f|A_i| = f(d^p - 1).$$

Specifying the α_w and $\beta_{\mathbf{u},w}$ variables. Next, we define the α_w and $\beta_{\mathbf{u},w}$. They are set differently based on three cases as follows.

Case 1. If $t \geq \left\lfloor \frac{n}{2} \right\rfloor$, then set $\beta_{\mathbf{u},w} = 1$ whenever $p+1 \leq j(w) \leq n-t-1$, and set all other α_w and $\beta_{\mathbf{u},w}$ to be 0. It can be verified straightforwardly that all constraints (DC-1) are satisfied. Their contribution to the dual objective value is

$$\sum_{\substack{\mathbf{u},w \\ p<j(w)<n-t}} \beta_{\mathbf{u},w} = (n-t-1-p)(d^{n-t} - d^{n-t-1}) - d^{n-t-1} + d^p.$$

Case 2. When $p+1 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor - 1$, set $\beta_{\mathbf{u},w} = 1$ for $p+1 \leq j(w) \leq t$, $\alpha_w = 1$ for $t+1 \leq j(w) \leq n-t-1$, and all other α_w and $\beta_{\mathbf{u},w}$ to be 0. All constraints (DC-1) are thus satisfied. The α_w and $\beta_{\mathbf{u},w}$ contribution to the objective is

$$\begin{aligned}
&\sum_{j=t+1}^{n-t-1} (d^{n-j} - d^{n-j-1}) + \sum_{\substack{\mathbf{u},w \\ p+1 \leq j(w) \leq t}} \beta_{\mathbf{u},w} \\
&= (t-p)(d^{n-t} - d^{n-t-1}) + d^{n+p-2t-1} - d^t.
\end{aligned}$$

Case 3. When $t \leq p$ (which is $\leq n-t-1$), set $\alpha_w = 1$ for $p+1 \leq j(w) \leq n-t-1$ and all the other α_w and $\beta_{\mathbf{u},w}$ to be zero. Again, the feasibility of the constraints (DC-1) is easy to verify. The contribution to the objective value is

$$\sum_{j=p+1}^{n-t-1} (d^{n-j} - d^{n-j-1}) = d^{n-p-1} - d^t.$$

Specifying the $\gamma_{\mathbf{u}}$ and $\delta_{\mathbf{v}}$ variables. When $q = n-t$, set $\beta_{\mathbf{v}} = 1$ for all $v \in \bigcup_{j=n-t}^{n-1} B_j$ and all $\gamma_{\mathbf{u}} = 0$. The dual-objective contribution in this case is

$$\sum_{v \in \bigcup_{j=n-t}^{n-1} B_j} \delta_{\mathbf{v}} = |\bigcup_{j=n-t}^{n-1} B_j| = d^t - k.$$

When $n-t-1 \leq q \leq n$, define $\delta_{\mathbf{v}} = 1$ for all $\mathbf{v} \in \bigcup_{j=q}^{n-1} B_j$, $\gamma_{\mathbf{u}} = 1$ for all $\mathbf{u}$ such that $n-q+1 \leq i(\mathbf{u}) \leq n-p-1$, and all other $\delta_{\mathbf{v}}$ and $\gamma_{\mathbf{u}}$ are set to be zero. Note that

$$\sum_{\mathbf{v} \in \bigcup_{j=q}^{n-1} B_j} \delta_{\mathbf{v}} = \left| \bigcup_{j=q}^{n-1} B_j \right| \leq \min\{d^t - k, k(d^{n-q} - 1)\}.$$

To see the last inequality, note that $\left| \bigcup_{j=q}^{n-1} B_j \right|$ counts the number of strings in $W_0 - B$ which share a prefix of length at least q with some string $\mathbf{b}^{(l)}, l \in [k]$. As $|W_0| = d^t$, the upper-bound $d^t - k$ for the number of such strings is trivial. On the other hand, the number of strings sharing a prefix of length at least q with a fixed string $\mathbf{b}^{(l)}$ is at most $d^{n-q} - 1$ (discounting $\mathbf{b}^{(l)}$ itself). Hence, we get the upper-bound $k(d^{n-q} - 1)$ via a simple application of the union bound.

Consequently, the total contribution of the $\gamma_{\mathbf{u}}$ and $\delta_{\mathbf{v}}$ to the dual-objective is at most

$$\begin{aligned}
&\sum_{i=n-q+1}^{n-p-1} |A_i| + \min\{d^t - k, k(d^{n-q} - 1)\} \\
&\qquad = d^{q-1} - d^p + \min\{d^t - k, k(d^{n-q} - 1)\}.
\end{aligned}$$

The feasibility of all the constraints (DC-2) are easy to verify. Define the "cost" $c(k,p,q)$ to be the total contribution of all variables to the dual-objective value. For convenience, we summarize the values of $c(k,p,q)$ in Figure 4.

Theorem IV.4. *The above family of solutions is feasible for the dual LP* (4) *with objective value equal to* $c(k,p,q)$. *Consequently, for the network* $\log_d(N,0,m)$ *to be WSNB under the window algorithm with window size* d^t, *it is sufficient that*

$$m \geq 1 + \max_{1 \leq k \leq \min(f, d^t)} \min_{p,q} c(k,p,q). \tag{5}$$

The objective value $c(k,p,q)$

For $t \geq \left\lfloor \frac{n}{2} \right\rfloor$ and $q = n - t$,

$$c(k,p,q) = f(d^p - 1) + (n-t-1-p)(d^{n-t} - d^{n-t-1}) - d^{n-t-1} + d^p + d^t - k.$$

For $t \geq \left\lfloor \frac{n}{2} \right\rfloor$ and $q > n - t$,

$$c(k,p,q) = f(d^p - 1) + (n-t-1-p)(d^{n-t} - d^{n-t-1}) - d^{n-t-1} + d^{q-1} + \min\{d^t - k, k(d^{n-q} - 1)\}.$$

For $p + 1 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor - 1$ and $q = n - t$

$$c(k,p,q) = f(d^p - 1) + (t-p)(d^{n-t} - d^{n-t-1}) + d^{n+p-2t-1} - k.$$

For $p + 1 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor - 1$ and $q > n - t$

$$c(k,p,q) = f(d^p - 1) + (t-p)(d^{n-t} - d^{n-t-1}) + d^{n+p-2t-1} - d^t + d^{q-1} - d^p + \min\{d^t - k, k(d^{n-q} - 1)\}.$$

For $t \leq p$ and $q = n - t$,

$$c(k,p,q) = f(d^p - 1) + d^{n-p-1} - k.$$

For $t \leq p$ and $q > n - t$,

$$c(k,p,q) = f(d^p - 1) + d^{n-p-1} - d^t + d^{q-1} - d^p + \min\{d^t - k, k(d^{n-q} - 1)\}.$$

Fig. 4. The dual objective value of the family of dual-feasible solutions.

C. Selecting the best dual-feasible solution

It is a very straightforward though somewhat analytically tedious task to derive the best possible sufficient condition using Theorem IV.4. The idea is, for a given $k \leq \min(f, d^t)$, we first choose $p = p_k, q = q_k$ so that $c(k, p_k, q_k)$ is as small as possible. Then, derive an upperbound $C(t,f) \geq \max_k c(k, p_k, q_k)$. The sufficient condition is $m \geq C + 1$.

Theorem IV.5. *The $\log_d(N, 0, m)$ network is nonblocking under the window algorithm with window size d^t if $m \geq 1 + C(t,f)$ where $C(t,f)$ is defined in Figure 5.*

Proof: Consider 5 cases in the definition of $C(t,f)$. Due to space limitation, we will only specify for each k how p_k and q_k are chosen. The straightforward calculus task of checking that $c(k, p_k, q_k) \leq C(t,f)$ is omitted.

Case 1: $t < \left\lfloor \frac{n}{2} \right\rfloor, r \leq n - 2t - 1$. For any k, choose $p_k = \left\lceil \frac{n-r}{2} - 1 \right\rceil$ and $q_k = n - t$.

Case 2: $t < \left\lfloor \frac{n}{2} \right\rfloor, r \geq n - 2t$. For any k, set $p_k = 0$ and $q_k = n - t$.

Case 3: $t \geq \left\lfloor \frac{n}{2} \right\rfloor, r \geq n - t$. This case is a little trickier analytically. Define $x = \lfloor \log_d k \rfloor$. We set p_k and q_k differently depending on how large x is, so that the inequality $c(k, p_k, q_k) \leq C(t,f)$ always holds.

If $0 \leq x \leq 2t - n - 2$, which can only hold when $t \geq \frac{n+1}{2}$, then set $q_k = \left\lfloor \frac{n+x}{2} \right\rfloor + 1$ and $p_k = 0$. If $x = 2t - n - 1$ and $k \leq d^{x+1} - d^x$, then set $q_k = \left\lfloor \frac{n+x}{2} \right\rfloor + 1 = t$ and $p_k = 0$. If $x = 2t - n - 1$ and $k \geq d^{x+1} - d^x + 1$, then set set $q_k = n - t$ and and $p_k = 0$. Finally, when $x \geq 2t - n$, we again set $q_k = n - t$ and $p_k = 0$.

Case 4: $t \geq \left\lfloor \frac{n}{2} \right\rfloor, r \leq \min(2t - n - 2, n - t - 1)$. Set $p_k = n - t - r - 1$ and $q_k = \left\lfloor \frac{n+x}{2} \right\rfloor + 1$.

Case 5: $t \geq \left\lfloor \frac{n}{2} \right\rfloor, 2t - n - 2 < r \leq n - t - 1$. Set $p_k = n - t - r - 1$ and $q_k = \left\lfloor \frac{n+x}{2} \right\rfloor + 1$.

■

D. Some quick consequences of Theorem IV.5

All we have to do is to plug in the parameters t and f and compute $1 + C(t,f)$ to get the following results.

Corollary IV.6 (Theorem 4 in [28])**.** *Let $r = \lfloor \log_d f \rfloor$. The network $\log_d(N, 0, m)$ is f-cast strictly non-blocking if*

$$m \geq f\left(d^{\left\lceil \frac{n-r}{2} \right\rceil - 1} - 1\right) + d^{n - \left\lceil \frac{n-r}{2} \right\rceil}.$$

Proof: This corresponds to the $t = 0$ case of the window algorithm, which becomes an SNB condition as noted earlier.

$$C(0,f) = f\left(d^{\left\lceil \frac{n-r}{2} \right\rceil - 1} - 1\right) + d^{n - \left\lceil \frac{n-r}{2} \right\rceil} - 1.$$

■

The following corollary took about 6 pages in [27] to be proved (in two theorems), using complicated combinatorial reasoning. The result is on the general multicast case, without the fanout restriction f. In our setting, we can simply set $f = N = d^n$. In fact, even though the corollary states exactly the same results as in [27], the statement is quite a bit simpler.

Corollary IV.7 (Theorems 1 and 2 in [27])**.** *The d-ary multi-log network $\log_d(N, 0, m)$ is WSNB with respect to the window algorithm with window size d^t if m is at least*

$$\begin{cases} d^{n-2t-1} + td^{n-t-1}(d-1), \ \textbf{when } t \leq \left\lfloor \frac{n}{2} \right\rfloor - 1, \\ d^{n-t-1}[(d-1)(n-t-1) - 1] + d^t - d^{2t-n-1}(d-1) + 1, \\ \quad \textbf{when } t \geq \left\lfloor \frac{n}{2} \right\rfloor. \end{cases}$$

Proof: Note that $r = n$ and $C(t, d^n) = d^{n-2t-1} + td^{n-t-1}(d-1) - 1$ when $t \leq \left\lfloor \frac{n}{2} \right\rfloor - 1$, and $C(t, d^n) = d^{n-t-1}[(d-1)(n-t-1)-1] + d^t - d^{2t-n-1}(d-1)$ otherwise.

■

The upper-bound $C(t,f)$

To shorten the notations, let $r = \lfloor \log_d f \rfloor$.

$$C(t,f) = \begin{cases} f\left(d^{\lceil \frac{n-r}{2} \rceil - 1} - 1\right) + d^{n - \lceil \frac{n-r}{2} \rceil} - 1 & t < \lfloor \frac{n}{2} \rfloor, r \leq n - 2t - 1 \\ t(d-1)d^{n-t-1} + d^{n-2t-1} - 1 & t < \lfloor \frac{n}{2} \rfloor, r \geq n - 2t \\ [(n-t-1)(d-1) - 1]d^{n-t-1} + d^t - (d-1)d^{2t-n-1} & t \geq \lfloor \frac{n}{2} \rfloor, r \geq n - t \\ f\left(d^{n-t-r-1} - 1\right) + [r(d-1) - 1]d^{n-t-1} + d^{\lfloor \frac{n+r}{2} \rfloor} + f\left(d^{n - \lfloor \frac{n+r}{2} \rfloor - 1} - 1\right) & t \geq \lfloor \frac{n}{2} \rfloor, r \leq \min(2t - n - 2, n - t - 1) \\ f\left(d^{n-t-r-1} - 1\right) + [r(d-1) - 1]d^{n-t-1} + d^t - (d-1)d^{2t-n-1} & t \geq \lfloor \frac{n}{2} \rfloor, 2t - n - 2 < r \leq n - t - 1 \end{cases}$$

Fig. 5. We show in Theorem IV.5 that $C(t,f) \geq \max_k \min_{p,q} c(k,p,q)$

V. Concluding Remarks

In the photonic-switching case, a common requirement is that of *crosstalk-free* as crosstalk between interfering channels is one of the major obstacles in designing cost-effective photonic-switches [20], [29]. We can prove the crosstalk free version of Theorem IV.5 using exactly the same method. We will not state the theorem due to space limitation. Just like in the case of IV.5, many known results are immediate consequences of the crosstalk free version.

References

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

[2] 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)*, Dec 1973, pp. 21–28.

[3] V. E. Beneš, *Mathematical theory of connecting networks and telephone traffic*. New York: Academic Press, 1965, mathematics in Science and Engineering, Vol. 17.

[4] F. K. Hwang, *The mathematical theory of nonblocking switching networks*. River Edge, NJ: World Scientific Publishing Co. Inc., 1998.

[5] J. S. Turner and R. Melen, "Multirate Clos networks," *Communications Magazine, IEEE*, vol. 41, no. 10, pp. 38–44, 2003.

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

[7] S. C. Liew, M.-H. Ng, and C. W. Chan, "Blocking and nonblocking multirate clos switching networks." *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 307–318, 1998.

[8] H. Q. Ngo and V. H. Vu, "Multirate rearrangeable Clos networks and a generalized bipartite graph edge coloring problem," *SIAM Journal on Computing*, vol. 32, no. 4, pp. 1040–1049, 2003.

[9] H. Q. Ngo, "A new routing algorithm for multirate rearrangeable Clos networks," *Theoret. Comput. Sci.*, vol. 290, no. 3, pp. 2157–2167, 2003.

[10] B. Gao and F. K. Hwang, "Wide-sense nonblocking for multirate 3-stage Clos networks," *Theoret. Comput. Sci.*, vol. 182, no. 1-2, pp. 171–182, 1997.

[11] H. Q. Ngo, D. Pan, and C. Qiao, "Constructions and analyses of nonblocking wdm switches based on arrayed waveguide grating and limited wavelength conversion," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 205–217, 2006.

[12] G. Wilfong, B. Mikkelsen, C. Doerr, and M. Zirngibl, "WDM cross-connect architectures with reduced complexity," *Journal of Lightwave Technology*, vol. 17, no. 10, pp. 1732–1741, Oct 1999.

[13] P. E. Haxell, A. Rasala, G. T. Wilfong, and P. Winkler, "Wide-sense nonblocking wdm cross-connects," in *Algorithms—ESA '03 (Rome, Italy)*, ser. Lecture Notes in Comput. Sci. Berlin: Springer, 2002, vol. 2461, pp. 538–549.

[14] A. Rasala and G. Wilfong, "Strictly non-blocking WDM cross-connects," in *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2000, San Francisco, CA)*. New York: ACM, 2000, pp. 606–615.

[15] ——, "Strictly non-blocking WDM cross-connects for heterogeneous networks," in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (STOC'2000, Portland, OR)*. New York: ACM, 2000, pp. 513–524.

[16] J. Duato, S. Yalamanchill, and L. Ni, *Interconnection Networks*. Washington: IEEE Computer Society, 1997.

[17] 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.

[18] M. M. Vaez and C.-T. Lea, "Strictly nonblocking directional-coupler-based switching networks under crosstalk constraint," *IEEE Trans. Comm.*, vol. 48, no. 2, pp. 316–323, Feb 2000.

[19] 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.

[20] G. Maier and A. Pattavina, "Design of photonic rearrangeable networks with zero first-order switching-element-crosstalk," *IEEE Trans. Comm.*, vol. 49, no. 7, pp. 1248–1279, Jul 2001.

[21] 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.

[22] H. Q. Ngo, Y. Wang, and A. Le, "A linear programming duality approach to analyzing strictly nonblocking d-ary multilog networks under general crosstalk constraints," in *Computing and combinatorics*, ser. Lecture Notes in Comput. Sci. Berlin: Springer, 2008, vol. 5092, pp. 510–520.

[23] E. G. Coffman, M. R. Garey, and D. S. Johnson, "Dynamic bin packing," *SIAM J. Comput.*, vol. 12, no. 2, pp. 227–258, 1983.

[24] Y. Wang, H. Q. Ngo, and X. Jiang, "Strictly nonblocking f-cast d-ary multilog networks under fanout and crosstalk constraints," in *Proceedings of the 2008 International Conference on Communications (ICC)*. Bejing, China: IEEE, 2008.

[25] K.-H. Tsai, D.-W. Wang, and F. Hwang, "Lower bounds for wide-sense nonblocking Clos network," *Theoret. Comput. Sci.*, vol. 261, no. 2, pp. 323–328, 2001, computing and combinatorics (Taipei, 1998).

[26] Y. Tscha and K.-H. Lee, "Yet another result on multi-$\log_2 n$ networks," *IEEE Transactions on Communications*, vol. 47, no. 9, pp. 1425–1431, Sep 1999.

[27] G. Danilewicz, "Wide-sense nonblocking $log_d(n,0,p)$ multicast switching networks," *IEEE Transactions on Communications*, vol. 55, no. 11, pp. 2193–2200, 2007.

[28] F. K. Hwang, Y. Wang, and J. Tan, "Strictly nonblocking f-cast $\log_d(n,m,p)$ networks," *IEEE Transactions on Communications*, vol. 55, no. 5, pp. 981–986, May 2007.

[29] M. Vaez and C.-T. Lea, "Wide-sense nonblocking Banyan-type switching systems based on directional couplers," *IEEE J. Select. Areas Commun.*, vol. 16, no. 7, pp. 1327–1332, Sep 1998.