A New Routing Algorithm for Multirate Rearrangeable Clos Networks

Hung Q. Ngo *

May 21, 2002

Abstract

In this paper, we study the problem of finding routing algorithms on the multirate rearrangeable Clos networks which use as few number of middle-stage switches as possible. We propose a new routing algorithm called the "grouping algorithm". This is a simple algorithm which uses fewer middle-stage switches than all known strategies, given that the number of input-stage switches and output-stage switches are relatively small compared to the size of input and output switches. In particular, the grouping algorithm implies that $m = 2n + \lceil \frac{n-1}{2^k} \rceil$ is a sufficient number of middle-stage switches for the symmetric 3-stage Clos network $C(n, m, r)$ to be multirate rearrangeable, where k is any positive integer and $r \leq n/(2^k - 1)$.

1 Introduction

The Clos network has been widely used for data communications and parallel computing systems. Quite a lot of research efforts [1, 3, 13, 11, 4, 12, 10, 5, 6, 8, 14, 2, 7] have been put on investigating the non-blocking properties and rearrangeability of the Clos network. The 3-stage Clos network was paid special attention to since it can be expanded in a "straightforward" way to multi-stage Clos network. Let us first formally introduce some related concepts.

The 3-state 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 switch I_i ($i = 1, \dots, r_1$) is connected to each middle switch M_j ($j = 1, \dots, m$). Similarly, the middle-stage switches M_j and third-stage swiches J_i are fully connected. The *symmetric* 3*-stage Clos network* $C(n, m, r)$ is nothing but $C(n, r, m, n, r)$. A $C(2, 3, 4)$ is shown in Figure 2. Any switch is assumed to be nonblocking, i.e. any inlet can be connected to any outlet as long as there's no conflict on the outlet. This can be thought of as a crossbar of size $p \times q$ with pq cross-points. Having too many cross-points is expensive and we would like to design a huge switch using smaller switches with fewer number of cross-points than when a brute-force design is used. The inlets (outlets) of the input (output) switches are the *inputs* (*outputs*) of the network. Inputs and outputs are referred to as *external links*, while links between switches are referred to as *internal links*.

In the multi-rate environment, a *connection request* is a triple (x, y, w) where x is an inlet, y an outlet, and $w \in (0, 1]$ the weight. A *request frame* is a collection of requests such that the total weight of all requests in the frame involving a fixed inlet or outlet does not exceed unity.

*Computer Science and Engineering Department, 201 Bell Hall State University of New York at Buffalo, Amherst, NY 14260, USA. hungngo@cse.buffalo.edu

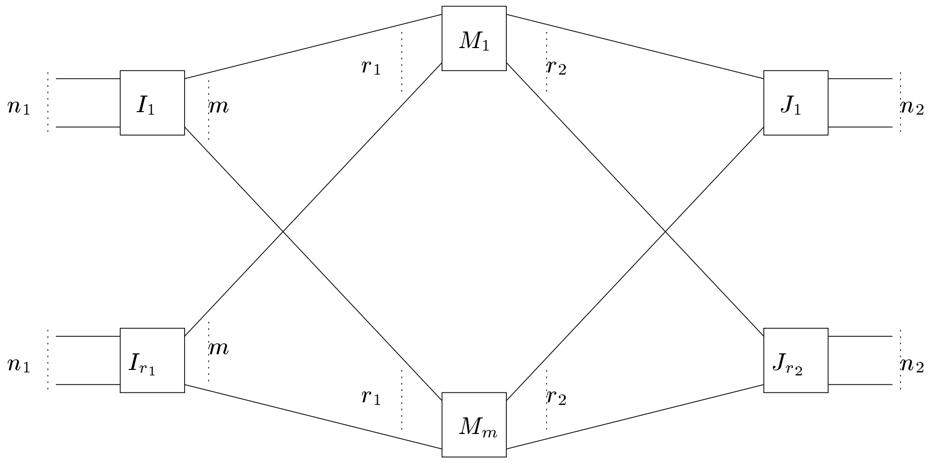

Figure 1: The 3-stage Clos network $C(n_1, r_1, m, n_2, r_2)$

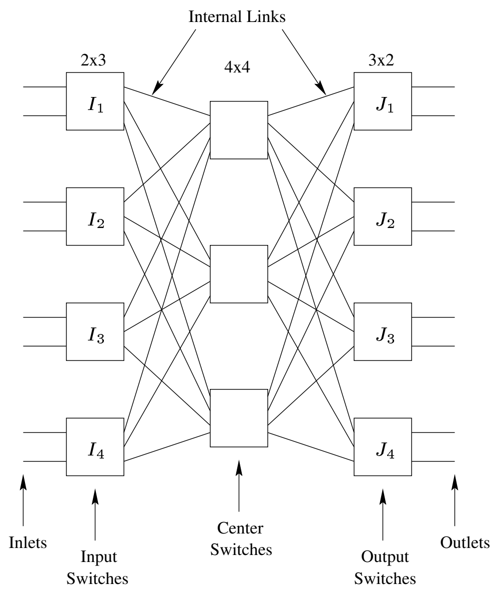

Figure 2: The symmetric 3-stage Clos network $C(2, 3, 4)$

This condition simply refers to the fact that each external link can carry a set of requests whose total rate is at most 1. To discuss routing it is convenient to assume that all links are directed from left to right. Thus a *path* from an inlet to any outlet always consists of the sequence: an inlet link $\rightarrow$ an input switch $\rightarrow$ a link $\rightarrow$ a center switch $\rightarrow$ a link $\rightarrow$ an output switch $\rightarrow$ an outlet link. Furthermore, since the crossbars are assumed to be nonblocking, a request (x, y, w) is *routable* if and only if there exists a path from x to y such that every link on this path has unused capacity at least $1 - w$ before carrying out this request. A request frame is routable if there exists a set of paths, one for each request, such that for every link the sum of weights of all requests going through it does not exceed unity. The Clos network $C(n_1, r_1, m, n_2, r_2)$ is said to be *multi-rate rearrangeable* (or just rearrangeable as in this paper we only consider the multi-rate environment) if *every* request frame is routable.

Let $m(n, r)$ denote the minimum value of m such that $C(n, m, r)$ is (multi-rate) rearrangeable for $n, r \geq 2$. (The cases where either n or r are 1 is trivial, hence we only consider $n, r \geq 2$ from here on.) Our problem is to find routing algorithms which use as few number of middle-stage switches m as possible; essentially providing an upper bound for the function $m(n, r)$. Melen and Turner initiated the study of multirate networks (1989, [13]). Chung and Ross [3] conjectured that $m(n, r) \leq 2n - 1$ when the weights are chosen from a discrete set of K weights. So far no one has been able to prove or disprove the conjecture. The conjecture seems to hold even in the continuous bandwidth case.

Let us preview some previous works on this problem.

Du et al. (1999, [4]) showed that

$$\lceil 11n/9 \rceil \leq m(n, r) \leq 41n/16 + O(1). \tag{1}$$

Lin et al. (1999, [11]) comfirmed Chung-Ross conjecture for a restricted discrete bandwidth case where each connection has a weight chosen from a set $\{1 \geq p_1 > \cdots > p_h > 1/2 \geq p_{h+1} > \cdots > p_k\}$ which satisfies the condidion that p_i is an integer multiple of p_{i+1} for $i = h + 1, \ldots, k - 1$.

Hu et al. (2001, [7]) studied the monotone routing strategy and showed that under this strategy

$$\begin{aligned} m(n, r) &\leq 2n + 1 \text{ for } n = 2, 3, 4 \quad (2) \\ m(n, r) &\leq 2n + 3 \text{ for } n = 5, 6. \quad (3) \end{aligned}$$

In this paper we propose a new routing algorithm called the *grouping algorithm* which route all requests whose weights are $> 1/2^k$ for any positive integer k. It is then shown that the grouping algorithm requires

$$\left\lfloor 2n + r - \frac{n + r}{2^{k-1}} \right\rfloor$$

middle-stage switches to route all these requests. Several consequences of this routing algorithm shall be derived. In particular, a consequence of the algorithm is that the symmetric 3-stage Clos network $C(n, 2n + \lceil \frac{n}{2^k} \rceil, r)$ is multirate rearrangeable if $r \leq n/(2^k - 1)$, for any positive integer k. This new bound beats all existing bounds on the minimum number of middle-stage switches.

2 Main Results

In this section, we first describe the grouping algorithm to route all requests with weights $> 1/2^k$, and then derive several consequences of the algorithm.

Let $\mathcal{F} = \{(x, y, w)\}$ be a set of connection requests. Recall that for each connection request (x, y, w), x is an inlet of some input switch I, y is an outlet of some output switch J, and $w \in (0, 1]$ is the weight of the request. For the request frame $\mathcal{F}$ to be valid, it must satisfy the condition that

$$\sum_{y:(x,y,w)\in\mathcal{F}} w \leq 1, \quad \text{for any inlet } x, \tag{4}$$

$$\sum_{x:(x,y,w)\in\mathcal{F}} w \leq 1, \quad \text{for any outlet } y. \tag{5}$$

Namely, the total weight of requests involving a fixed inlet x or a fixed outlet y does not exceed unity. We use $\mathcal{I}$ and $\mathcal{J}$ to denote the set of input switches and output switches, respectively. Note that $|\mathcal{I}| = |\mathcal{J}| = r$, as we are considering the symmetric 3-stage Clos network $C(n, m, r)$. Let $(x, y, w) \in \mathcal{F}$ be a connection request. If x is an inlet of input switch $I \in \mathcal{I}$ and y is an outlet of output switch $J \in \mathcal{J}$, then we refer to the request (x, y, w) as an (I, J)-request of weight w. Beside the conditions (4) and (5), the fact that a connection request is from a particular inlet or to a particular outlet is immaterial as far as routing is concerned. A middle-stage switch M can carry a set of connection requests as long as the total weight of requests that M carries which involve a particular input switch I or a particular output switch J does not exceed unity. It does not matter which inlets of I the requests are from, nor which outlets of J the requests are to.

Let k be a fixed positive integer. For each $l = 1, \ldots, k$, $I \in \mathcal{I}$ and $J \in \mathcal{J}$, let $S(I, J, l)$ be the set of (I, J)-requests in $\mathcal{F}$ whose weights are in the interval $(\frac{1}{2^l}, \frac{1}{2^{l-1}}]$. Clearly the set of requests with weights $> 1/2^k$ is the union of $S(I, J, l)$ over all $I \in \mathcal{I}$, $J \in \mathcal{J}$, and $l = 1, \ldots, k$. We are now ready to describe the grouping algorithm.

Algorithm 2.1 (The Grouping Algorithm). Let $S(I, J, l)$ be the set of requests initialized as above.

for all pairs $(I, J) \in \mathcal{I} \times \mathcal{J}$ **do**
    for $l = k$ down to 2 **do**
        while $|S(I, J, l)| \geq 2$ **do**
            Let w_1 and w_2 be any two weights in $S(I, J, l)$.
            Remove w_1 and w_2 from $S(I, J, l)$.
            Create a new weight $w = w_1 + w_2$
            // note that $w \in (\frac{1}{2^{l-1}}, \frac{1}{2^{l-2}}]$
            Add a new (I, J)-request with weight w into $S(I, J, l-1)$
        end while
    end for
    // At this point, each set $S(I, J, l)$, $l \geq 2$, has at most one (I, J)-request left
    if $|S(I, J, l)| = 1$ for some $l \geq 2$ **then**
        Create a new (I, J)-request with weight w_{IJ} equal to the total weight of all requests in $S(I, J, l)$, $l = 2, \ldots, k$.
        // We remove the requests in $S(I, J, l)$, $l \geq 2$, later for convinience
    end if
end for
Now, remove all requests in $S(I, J, l)$, $l \geq 2$, as the w_{IJ} cover these requests
// at this point, for each pair (I, J) there are only requests
// in $S(I, J, 1)$ and possibly an extra request with weight w_{IJ}
Route all requests in the sets $S(I, J, 1)$ and the extra w_{IJ} as if the network is in the classical environment.

Some explanation is in order. Suppose in the request frame $\mathcal{F}$, there are two (I, J)-requests with weight w_1 and w_2 where $w = w_1 + w_2 \leq 1$. Remove w_1 and w_2 from $\mathcal{F}$, add to $\mathcal{F}$ a new (I, J)-request with weight w. Then, any valid routing of the new $\mathcal{F}$ induces a valid routing of the old $\mathcal{F}$. Basically, if the new (I, J)-request is routed through middle-stage switch M, then we route both of the w_1-request and w_2-request through M. This is the idea behind lines 4-9 of the algorithm.

Secondly, we need to explain what we mean on line 20. The classical environment is the environment where each middle switch can carry at most one request from each input switch I and at most one request to each output switch J. If m is the maximum number of requests involving an input switch or an output switch, then m middle-stage switches is necessary and sufficient to route all requests. This is a consequence of the König's Line Coloring Theorem [9]. For another proof using P. Hall's matching condition, see [1]. For example, when there are at most n requests out of each input switch or to each output switch, n middle-stage switches is sufficient. This is the celebrated Slepian-Duguid theorem [1].

Consequently, in order to determine how many middle-stage switches the Grouping Algorithm requires, we only need to determine the maximum number of requests out of an input switch I or to an output switch J.

We first formally show the correctness of the Grouping Algorithm

Lemma 2.2. *The Grouping Algorithm correctly route all requests with weights* $> 1/2^k$.

Proof. The part from line 1 to line 10 is clear from the observation made above. The new weight w is at most $1/2^{l-2}$, as noted in line 7 of the algorithm. As $l \geq 2$, we have $w \leq 1$. We only need to show that w_{IJ}, if created, is also at most 1. But, as noted on line 11, the new weight w_{IJ} is at most

$$\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{k-1}} < 1.$$

$\square$

Remark 2.3. We are not concerned so much with the running time of our algorithm. The main objective of this paper is to find a new bound for the number of middle-stage switches for $C(m, n, r)$ to be multirate rearrangeable. The version given in Algorithm 2.1 can be significantly improved in terms of running time. However, we gave the simple version so that the proofs are easier to follow.

To see how the running time can be improved, notice that we could have, for each $l \geq 2$, combined every 2^p weights in $S(I, J, l)$ to form a new weight in $S(I, J, l - p)$. Thus, this combination can be done "simultaneously" by writing the size $|S(I, J, l)|$ in binary format. We leave the rest of the procedure to the reader.

To this end, we seek the maximum numer of middle-stage switches used by this algorithm. This is done by observing several facts, formally put in the following lemmas.

Lemma 2.4. *Let the sets* $S(I, J, l)$ *be defined as before the Grouping Algorithm is run. Then, for each* $I \in \mathcal{I}$

$$\sum_{J \in \mathcal{J}} \left(\frac{|S(I, J, 1)|}{2^0} + \frac{|S(I, J, 2)|}{2^1} + \cdots + \frac{|S(I, J, k)|}{2^{k-1}} \right) \leq 2n - \frac{n}{2^{k-1}}. \tag{6}$$

Similarly, for each $J \in \mathcal{J}$, *we have*

$$\sum_{I \in \mathcal{I}} \left(\frac{|S(I, J, 1)|}{2^0} + \frac{|S(I, J, 2)|}{2^1} + \cdots + \frac{|S(I, J, k)|}{2^{k-1}} \right) \leq 2n - \frac{n}{2^{k-1}}. \tag{7}$$

Proof. We show inequality (6). Inequality (7) is obtained in a completely similar fashion. Consider a request frame $\mathcal{F} = \{(x, y, w)\}$. For each inlet x of an input-switch I, and each $l = 1, \ldots, k$, let $s(x, l)$ be the number of requests $(x, y, w) \in \mathcal{F}$ where $w \in (\frac{1}{2^l}, \frac{1}{2^{l-1}}]$. Inequality (4) implies

$$\begin{aligned} 1 &\geq \sum_{y:(x,y,w)\in\mathcal{F}} w \\ &> s(x,1)\frac{1}{2} + s(x,2)\frac{1}{4} + \cdots + s(x,k)\frac{1}{2^k}. \end{aligned}$$

Hence,

$$2^k > 2^{k-1}s(x,1) + 2^{k-2}s(x,2) + \cdots + 2^0 s(x,k).$$

Consequently, as all of the numbers $s(x, l)$ are integers, we must have

$$2^k - 1 \geq 2^{k-1}s(x,1) + 2^{k-2}s(x,2) + \cdots + 2^0 s(x,k). \tag{8}$$

Notice that

$$\sum_{J\in\mathcal{J}} |S(I,J,l)| = \sum_{x=1}^{n} s(x,l).$$

In words, the total number of requests involving I with weights in the interval $(\frac{1}{2^l}, \frac{1}{2^{l-1}}]$ is the sum over all inlets x of I of the number of requests involving x with weights in the same interval. Thus, summing inequality (8) over all n inlets x of I, we obtain

$$\begin{aligned} n(2^k - 1) &\geq \sum_{x=1}^{n} \Big(2^{k-1}s(x,1) + 2^{k-2}s(x,2) + \cdots + 2^0 s(x,k)\Big) \\ &= 2^{k-1}\sum_{x=1}^{n} s(x,1) + 2^{k-2}\sum_{x=1}^{n} s(x,2) + \cdots + 2^0 \sum_{x=1}^{n} s(x,k) \\ &= 2^{k-1}\sum_{J\in\mathcal{J}} |S(I,J,1)| + 2^{k-2}\sum_{J\in\mathcal{J}} |S(I,J,2)| + \cdots + 2^0 \sum_{J\in\mathcal{J}} |S(I,J,k)|. \end{aligned}$$

Dividing both sides of this inequality by 2^{k-1} yields (6). □

Lemma 2.5. *During the execution from lines* 1 *to lines* 16 *of Algorithm 2.1, for each input-switch* I *the following sum is invariant:*

$$\sum_{J\in\mathcal{J}} \left(\frac{|S(I,J,1)|}{2^0} + \frac{|S(I,J,2)|}{2^1} + \cdots + \frac{|S(I,J,k)|}{2^{k-1}} \right).$$

Similarly, for each $J \in \mathcal{J}$, *the following sum is unchanged:*

$$\sum_{I\in\mathcal{I}} \left(\frac{|S(I,J,1)|}{2^0} + \frac{|S(I,J,2)|}{2^1} + \cdots + \frac{|S(I,J,k)|}{2^{k-1}} \right).$$

Proof. At any particular value of l, from line 4 to line 8 of the algorithm, we increase $|S(I, J, l - 1)$ by one and decrease $|S(I, J, l)|$ by two. But,

$$\frac{|S(I,J,l-1)|}{2^{l-2}} + \frac{|S(I,J,l)|}{2^{l-1}} = \frac{|S(I,J,l-1)|+1}{2^{l-2}} + \frac{|S(I,J,l)|-2}{2^{l-1}}.$$

Hence, both of the sums are not changed as desired. □

Lemma 2.6. *Right before line* 20 *of the Grouping Algorithm, the total number of requests involving a particular input-switch I or an output-switch J is at most*

$$2n + r - \frac{n+r}{2^{k-1}}.$$

Proof. By the previous two lemmas, right before line 17 inequalities (6) and (7) still hold. Consider any input-switch I. Right before line 20, the number of requests $r(I)$ involving I is at most

$$\sum_{J \in \mathcal{J}} \left(|S(I,J,1)| + f(I,J)\right),$$

where

$$f(I,J) = \begin{cases} 1 & \text{if } w_{IJ} \text{ was created} \\ 0 & \text{otherwise} \end{cases}.$$

Hence,

$$\begin{aligned} r(I) &\leq \sum_{J \in \mathcal{J}} \left(|S(I,J,1)| + f(I,J)\right) \\ &= \sum_{J \in \mathcal{J}} \left(\frac{|S(I,J,1)|}{2^0} + \frac{|S(I,J,2)|}{2^1} + \cdots + \frac{|S(I,J,k)|}{2^{k-1}}\right) + \\ &\quad \sum_{J \in \mathcal{J}} \left(f(I,J) - \frac{|S(I,J,2)|}{2^1} - \cdots - \frac{|S(I,J,k)|}{2^{k-1}}\right). \end{aligned}$$

By the definition of $f(I,J)$, it is 1 when at least one of $|S(I,J,l)|$ is 1. Thus,

$$f(I,J) - \frac{|S(I,J,2)|}{2^1} - \cdots - \frac{|S(I,J,k)|}{2^{k-1}} \leq 1 - \frac{1}{2^{k-1}}.$$

This fact and inequality (6) give

$$\begin{aligned} r(I) &\leq \sum_{J \in \mathcal{J}} \left(\frac{|S(I,J,1)|}{2^0} + \frac{|S(I,J,2)|}{2^1} + \cdots + \frac{|S(I,J,k)|}{2^{k-1}}\right) + \\ &\quad \sum_{J \in \mathcal{J}} \left(f(I,J) - \frac{|S(I,J,2)|}{2^1} - \cdots - \frac{|S(I,J,k)|}{2^{k-1}}\right). \\ &\leq 2n - \frac{n}{2^{k-1}} + \sum_{J \in \mathcal{J}} \left(1 - \frac{1}{2^{k-1}}\right) \\ &= 2n + r - \frac{n+r}{2^{k-1}}. \end{aligned}$$

The fact that the number of requests involving an output-switch J is also at most this number is shown similarly. □

Theorem 2.7. *The Grouping Algorithm requires at most*

$$\left\lfloor 2n + r - \frac{n+r}{2^{k-1}} \right\rfloor$$

middle-stage switches to route all requests with weights $> 1/2^k$, for any positive integer k.

Proof. Recall the observation made in the paragraphs right before Lemma 2.2. The number of middle-stage switches required is at most the number of requests involving an input-switch or an output-switch. Since this number of an integer, this theorem follows immediately from the previous one. □

Several consequences of this algorithm can now be derived.

Corollary 2.8. *The Grouping Algorithm can route all requests (not just the ones $> 1/2^k$) using at most $2n - 1 + r$ middle-stage switches. In other words, the Clos network $C(n, 2n - 1 + r, r)$ is multirate rearrangeable.*

Proof. Let k be large enough so that all requests have weights $> 1/2^k$. The Grouping Algorithm then routes all requests. This follows directly from the Theorem 2.7. □

Note that for r relatively small compared to n, Corollary 2.8 is better than all previously known results as introduced in the first section. For example, when $r \leq n/2$, the Clos network $C(n, m, r)$ is multirate rearrangeable with $5n/2 + O(1)$ number of middle-stage switches. When $r \leq n/4$, we need only $9n/4 + O(1)$, and so on. However, combining Theorem 2.7 with a lemma of Du et al. [4] we are able to do even better, as formally put in the following corollary.

Corollary 2.9. *The Clos network $C(n, 2n + \lceil \frac{n-1}{2^{k-1}} \rceil, r)$ is multirate rearrangeable, given that $r \leq \frac{n}{2^{k-1}-1}$. For any positive integer $k \geq 2$.*

Proof. Lemma 3 in [4] essentially states that if $c \geq 2n$ number of middle switches are sufficient to route all requests with weights $> 1/f$, where f is a positive integer, then at most

$$\max\{\lceil (c-2)/f - c + 2n \rceil, 0\}$$

more middle-stage switches are needed to route all the rest of the requests.

Now assume $r \leq \frac{n}{2^{k-1}-1}$, then Theorem 2.7 says that at most

$$\left\lfloor 2n + r - \frac{n+r}{2^{k-1}} \right\rfloor \leq 2n$$

middle-stage switches are sufficient to route all requests with weights $> 1/2^k$. Add more empty middle-stage switches so that we have precisely $c = 2n$ middle-stage switches, then apply Lemma 3 of [4] with $f = 1/2^k$, we get the total number of middle-stage switches needed is at most

$$2n + \lceil (2n-2)/2^k \rceil,$$

which is the desired result. □

A few special cases of the previous corollary for $k = 2, 3, 4$ illustrate the fact that this new result is better than the old bound of $41n/16 + O(1)$ in [4].

Corollary 2.10. *We have*

(i) $C(n, \lceil \frac{5n-1}{2} \rceil, r)$ *is multirate rearrangeable when* $r \leq n$.

(ii) $C(n, \lceil \frac{9n-1}{4} \rceil, r)$ *is multirate rearrangeable when* $r \leq n/3$.

(iii) $C(n, \lceil \frac{17n-1}{8} \rceil, r)$ *is multirate rearrangeable when* $r \leq n/7$.

3 Discussions

In this paper, we proposed a routing algorithm for the multirate rearrangeable symmetric 3-stage Clos network $C(n, m, r)$. Several nice consequences of the routing algorithm were derived. In particular, the minimum number of middle-stage switches is better than all existing results, given that r is relatively small compared to n. Our result, although still restricted, is more general than the improvement made in [7], for example. Admittedly though, our algorithm is more complicated and is slower than the monotone routing algorithm introduced in [7].

It should be noted that the results in the paper do not have to be restricted to symmetric Clos network $C(n, m, r)$. The symmetric case was presented for simplicity. In the general $C(n_1, r_1, m, n_2, r_2)$, if we let $n = \max\{n_1, n_2\}$, and $r = \max\{r_1, r_2\}$, then the Grouping Algorithm still works, Theorem 2.7 and its consequences still hold in a completely similar fashion.

References

[1] V. E. BENEŠ, *Mathematical theory of connecting networks and telephone traffic*, Academic Press, New York, 1965. Mathematics in Science and Engineering, Vol. 17.

[2] J. D. CARPINELLI AND A. Y. ORUÇ, *Applications of matching and edge-coloring algorithms to routing in Clos networks*, Networks, 24 (1994), pp. 319–326.

[3] S.-P. CHUNG AND K. W. ROSS, *On nonblocking multirate interconnection networks*, SIAM J. Comput., 20 (1991), pp. 726–736.

[4] D. Z. DU, B. GAO, F. K. HWANG, AND J. H. KIM, *On multirate rearrangeable Clos networks*, SIAM J. Comput., 28 (1999), pp. 464–471 (electronic).

[5] P. FISHBURN, F. K. HWANG, D. Z. DU, AND B. GAO, *On 1-rate wide-sense nonblocking for 3-stage Clos networks*, Discrete Appl. Math., 78 (1997), pp. 75–87.

[6] B. GAO AND F. K. HWANG, *Wide-sense nonblocking for multirate 3-stage Clos networks*, Theoret. Comput. Sci., 182 (1997), pp. 171–182.

[7] X.-D. HU, X.-H. JIA, D.-Z. DU, AND F. K. HWANG, *Monotone routing in multirate rearrangeable clos networks*, J. Parallel Distrib. Comput., 61 (2001), pp. 1382–1388.

[8] F. K. HWANG, *Rearrangeability of multiconnection three-stage Clos networks*, Networks, 2 (1972), pp. 301–306.

[9] D. KÖNIG, *Über graphen und ihre anwendung auf determinantentheorie und mengenlehre*, Math. Ann., 77 (1916), pp. 453–465.

[10] T. T. LEE AND P. P. TO, *Non-blocking routing properties of Clos networks*, in Advances in switching networks (Princeton, NJ, 1997), Amer. Math. Soc., Providence, RI, 1998, pp. 181–195.

[11] G.-H. LIN, D.-Z. DU, X.-D. HU, AND G. XUE, *On rearrangeability of multirate Clos networks*, SIAM J. Comput., 28 (1999), pp. 1225–1231 (electronic).

[12] G.-H. LIN, D.-Z. DU, W. WU, AND K. YOO, *On 3-rate rearrangeability of Clos networks*, in Advances in switching networks (Princeton, NJ, 1997), Amer. Math. Soc., Providence, RI, 1998, pp. 315–333.

[13] R. MELEN AND J. S. TURNER, *Nonblocking multirate networks*, SIAM J. Comput., 18 (1989), pp. 301–313.

[14] K.-H. TSAI AND D.-W. WANG, *Lower bounds for wide-sense non-blocking Clos network*, in Computing and combinatorics (Taipei, 1998), Springer, Berlin, 1998, pp. 213–218.