Theoretical Computer Science 411 (2010) 1669–1684

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

All-to-all personalized exchange in generalized shuffle-exchange networks☆

Well Y. Chou, Chiuyuan Chen*

Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 8 May 2009
Accepted 25 October 2009
Communicated by D.-Z. Du

Keywords:
Multistage interconnection networks
Shuffle-exchange networks
Omega network
Parallel and distributed computing
All-to-all communication
All-to-all personalized exchange

ABSTRACT

An all-to-all communication algorithm is said to be optimal if it has the smallest communication delay. Previous all-to-all personalized exchange algorithms are mainly for hypercube, mesh, and torus. In Yang and Wang (2000) [13], Yang and Wang proved that a multistage interconnection network (MIN) is a better choice for implementing all-to-all personalized exchange and they proposed optimal all-to-all personalized exchange algorithms for MINs. In Massini (2003) [9], Massini proposed a new optimal algorithm for MINs, which is independent of the network topology. Do notice that the algorithms in [9] and [13] work only for MINs with the unique path property (meaning that there is a unique path between each pair of source and destination) and satisfying $N = 2^n$, in which N is the number of processors, 2 means all the switches are of size 2×2, and n is the number of stages. In Padmanabhan (1991) [10], Padmanabhan proposed the generalized shuffle-exchange network (GSEN), which is a generalization of the shuffle-exchange network. Since a GSEN does not have the unique path property, the algorithms in [9] and [13] cannot be used. The purpose of this paper is to consider the all-to-all personalized exchange problem in GSENs. An optimal algorithm and several bounds will be proposed.

1. Introduction

Processors in a parallel and distributed processing system often need to communicate with other processors. The communication among these processors could be *one-to-one*, *one-to-many*, or *all-to-all*. All-to-all communication can be further classified into *all-to-all broadcast* and *all-to-all personalized exchange*. In all-to-all broadcast, each processor sends the same message to all other processors; while in all-to-all personalized exchange, each processor sends a specific message to every other processor. All-to-all personalized exchange occurs in many important applications (for example, matrix transposition and fast Fourier transform (FFT)) in parallel and distributed computing. The all-to-all personalized exchange problem has been extensively studied for hypercubes, meshes, and tori; see [9,13] for details. Although the algorithm for a hypercube achieves optimal time complexity, a hypercube suffers from unbounded node degrees and therefore has poor scalability; on the other hand, although a mesh or torus has a constant node degree and better scalability, its algorithm has a higher time complexity. In [13], Yang and Wang had proven that a multistage interconnection network (MIN) is a better choice for implementing all-to-all personalized exchange due to its shorter communication delay and better scalability.

Given N processors $P_0, P_1, \ldots, P_{N-1}$, an $N \times N$ MIN can be used in communication among these processors as shown in Figs. 1 and 2, where $N \times N$ means this MIN has N inputs and N outputs. A column in a MIN is called a *stage* and the nodes stages of a MIN are called *switches* (or *switching elements* or *crossbars*). Throughout this paper, N denotes the number of

☆ This research was partially supported by the National Science Council of the Republic of China under grant NSC97-2628-M-009-006-MY3.
* Corresponding author. Tel.: +886 3 5731767.
E-mail addresses: well.am94g@nctu.edu.tw (W.Y. Chou), cychen@mail.nctu.edu.tw, cychen@cc.nctu.edu.tw (C. Chen).

0304-3975/$ – see front matter © 2010 Published by Elsevier B.V.
doi:10.1016/j.tcs.2009.10.026

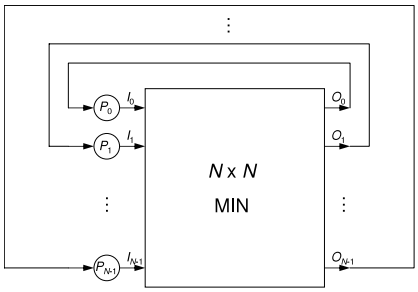

Fig. 1. Communications among processors using a MIN.

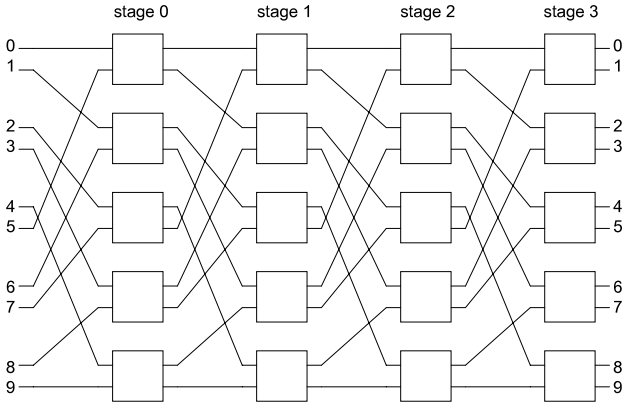

Fig. 2. A 10 × 10 MIN which is also a 10 × 10 GSEN.

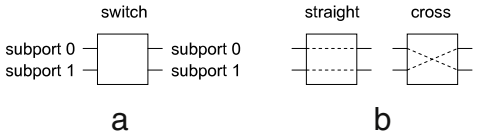

Fig. 3. (a) A 2 × 2 switch and its subports. (b) The two possible states of a 2 × 2 switch.

inputs (outputs) and n denotes the number of stages. Also, all the switches in a MIN are assumed to be of size 2×2. It is well known that a 2×2 switch has only two possible states: *straight* and *cross*, as shown in Fig. 3. A shuffle-exchange network (SEN) is also called an omega network (see [7]) and has been proposed as a popular architecture for MINs; see [3,6,10,12]. Since a SEN must satisfy $N = 2^n$, in [10], Padmanabhan generalized it to allow $N \neq 2^n$. More precisely, let N be an even integer. An $N \times N$ *generalized shuffle-exchange network* (GSEN) is a $\lceil \log_2 N \rceil$-stage $N \times N$ MIN such that each stage consists of the perfect shuffle on N terminals followed by $N/2$ switches. The N terminals in an $N \times N$ GSEN are numbered $0, 1, \ldots, N-1$ and the *perfect shuffle operation* on the N terminals is the permutation π defined by $\pi(i) = (2 \cdot i + \lfloor \frac{2 \cdot i}{N} \rfloor) \bmod N, \ \ 0 \leq i < N$. See Fig. 2 for an example. In [1,2], bidirectional GSENs are considered.

In the remaining discussion, unless otherwise specified, a MIN means an $N \times N$ MIN and a GSEN means an $N \times N$ GSEN. Do notice that we will follow the convention used in [1,2,10] that a GSEN has exactly $\lceil \log_2 N \rceil$ stages; $\lceil \log_2 N \rceil$ is the minimum number of stages to ensure that each input can get to each output. Based on this convention and for convenience, we will define

$$n = \lceil \log_2 N \rceil.$$

Clearly, for a GSEN, its N satisfies $2^{n-1} < N \leq 2^n$.

In this paper, an all-to-all communication algorithm is said to be *optimal* if it has the smallest communication delay. Now we review previous results. Yang and Wang [13] first considered the all-to-all personalized exchange problem for MINs. In particular, they proposed optimal all-to-all personalized exchange algorithms for a class of unique path, self-routable MINs; for example, baseline, omega, banyan networks, and the reverse networks of these networks. Note that a MIN is *unique path* if there is a unique path between each pair of source and destination and *self-routable* if the routing decision at a switch depends only on the addresses of the source and the destination. The algorithms in [13] can use the *stage control* technique (see [11]), which is a commonly used technique to reduce the cost of the network setting for all-to-all personalized exchange communication. Stage control means that the states of all the switches of a stage have to be identical. With stage control, a single control bit (0 for straight and 1 for cross), or in other words, one electronic driver circuit, can be used to control all the switches of a stage. Thus the number of expensive electronic driver circuits needed is significantly lower than that

of individual switch control. It was pointed out by Massini in [9] that the algorithms in [13] depend on network topologies and require pre-computation and memory allocation for Latin squares. In the same paper, Massini proposed a new optimal algorithm, which is independent on the network topology and does not require pre-computation or memory allocation for a Latin square. In [8], Liu et al. further generalized Massini's algorithm to MINs with $d \times d$ switches. See also [14].

When $N \neq 2^n$, it is possible to implement an $N \times N$ GSEN by using a $2^n \times 2^n$ SEN (recall that $2^{n-1} < N \leq 2^n$). For example, it is possible to implement a 514×514 GSEN by using a 1024×1024 SEN. A 514×514 GSEN uses 2570 switches while its corresponding 1024×1024 SEN uses 5120 switches; the former saves about 50% switches than the latter. To compare the hardware costs of a GSEN and a SEN, we calculate the numbers of switches used by an $N \times N$ GSEN and by its corresponding $2^n \times 2^n$ SEN for $N = 4, 6, 8, \ldots, 10002$. Among these 5000 N's,

- for 4175 (about 84%) out of them, a GSEN saves at least 10% switches than its corresponding SEN;
- for 3356 (about 67%) out of them, a GSEN saves at least 20% switches than its corresponding SEN;
- for 2537 (about 51%) out of them, a GSEN saves at least 30% switches than its corresponding SEN;
- for 1632 (about 33%) out of them, a GSEN saves at least 40% switches than its corresponding SEN.

Therefore a GSEN outperforms a SEN in hardware cost.

Do notice that although the algorithms in [9] and [13] are optimal, they work only for MINs that have the unique path property and satisfy $N = 2^n$. Since a GSEN is not a unique path MIN, the algorithms in [9] and [13] cannot be used. To our knowledge, no one has studied the all-to-all personalized exchange problem for MINs which do not have the unique path property and do not satisfy $N = 2^n$. The purpose of this paper is to consider the all-to-all personalized exchange problem for GSENs. In particular, we propose an optimal all-to-all personalized exchange algorithm for GSENs. This algorithm works for all N with $N \equiv 2 \pmod 4$. Let $\mathcal{R}(N)$ and $\mathcal{R}_{sc}(N)$ denote the minimum number of network configurations (defined in the next section) required to fulfill an all-to-all communication in a GSEN when the stage control technique is not assumed and assumed, respectively. Do notice that $\mathcal{R}(N)$ and $\mathcal{R}_{sc}(N)$ are closely related to the smallest communication delay. In particular, for a GSEN, the smallest communication delay of any all-to-all communication algorithm is $\theta(\mathcal{R}(N) + \log_2 N)$ and $\theta(\mathcal{R}_{sc}(N) + \log_2 N)$ when the stage control technique is not assumed and assumed, respectively. The optimal algorithms in [9] and [13] imply that $\mathcal{R}(2^n) = \mathcal{R}_{sc}(2^n) = 2^n$. In this paper, we will prove that, for $2^{n-1} < N \leq 2^n$, the followings hold:

- $N \leq \mathcal{R}(N) \leq \mathcal{R}_{sc}(N) \leq 2^n$;
- $\mathcal{R}_{sc}(N) = 2^n$;
- $\mathcal{R}(N) = N$ if $N \equiv 2 \pmod 4$;
- $\mathcal{R}(N) = 2^n$ if $k \geq 2, N \equiv 0 \pmod{2^k}, N \not\equiv 0 \pmod{2^{k+1}}$, and $2^{n-1} + 2^{n-k} \leq N \leq 2^n$;
- $\mathcal{R}(20) = 24$.

This paper is organized as follows: In Section 2, we give some preliminaries. In Section 3, we prove $N \leq \mathcal{R}(N) \leq \mathcal{R}_{sc}(N) = 2^n$. In Section 4, we propose an optimal all-to-all personalized exchange algorithm for GSENs with $N \equiv 2 \pmod 4$ and prove that $\mathcal{R}(N) = N$ if $N \equiv 2 \pmod 4$. In Section 5, we focus on GSENs with $N \equiv 0 \pmod 4$ and obtain several bounds. Some discussions and concluding remarks are given in the final section.

2. Preliminaries

In a GSEN, the switches are aligned in n stages: stage 0, stage 1, ..., stage $n-1$, with each stage consists of $N/2$ switches. The *network configuration* of a GSEN is defined by the states of its switches. Since a GSEN has $(N/2) \times n$ switches, its network configuration can be represented by an $(N/2) \times n$ matrix in which each entry is defined by the state of its corresponding switch. For example, the network configuration of the GSEN in Fig. 4(a) is shown in Fig. 4(b).

A *permutation* of a MIN is one-to-one mapping between the inputs and outputs. For a MIN, if there is a permutation that maps input i to output $p(i)$, where $p(i) \in \{0, 1, \ldots, N-1\}$ for $i = 0, 1, \ldots, N-1$, then we will use

$$\begin{pmatrix} 0 & 1 & \cdots & N-1 \\ p(0) & p(1) & \cdots & p(N-1) \end{pmatrix}$$

or simply

$$p(0)\, p(1)\, \cdots\, p(N-1)$$

to denote the permutation. Given the network configuration of a MIN, a permutation between the inputs and outputs can be obtained. For example, the network configuration shown in Fig. 4(a) maps input 0 to output 9, input 1 to output 7, input 2 to output 5, ..., and input 9 to output 0; thus this network configuration obtains the permutation 9 7 5 3 8 1 6 4 2 0.

The following conventions are used in the remaining part of this paper. Terminal i (j) is assumed on the left-hand (right-hand) side of the network and therefore is an input (output) processor. An (i, j)-*request* denotes a request for sending a message from i to j. An (i, j)-*path* denotes a path between i and j. Obviously, an (i, j)-request can be fulfilled by an (i, j)-path.

Consider an (i, j)-request and an (i, j)-path and see Fig. 5 for an illustration. An (i, j)-path P can be described by a sequence of labels that label the successive links on this path; a number whose binary representation corresponds to such a sequence is called a *control tag* or *tag* or *path descriptor* [1,2,4,10]. A control tag can be used as a header for routing a message: each successive switch uses the first element in the binary representation of the control tag to route the message, and then

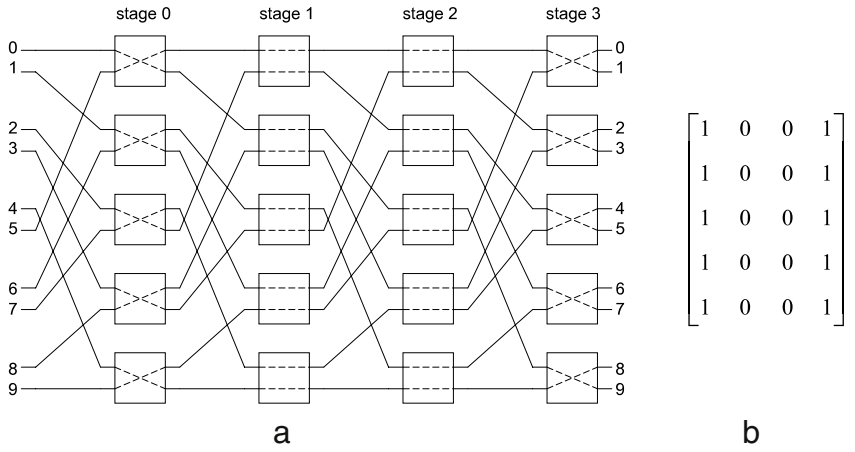

Fig. 4. (a) A 10 × 10 GSEN in which stage control is used. (b) The network configuration of the GSEN in (a).

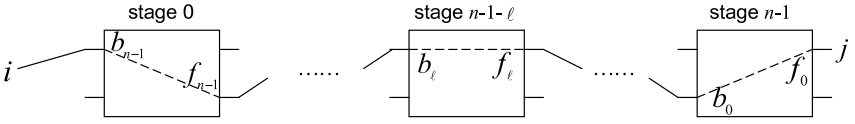

Fig. 5. An (i, j)-path P and the subports on P.

discards it. Take Fig. 4(a) for an example. Then $i = 2$ can get to $j = 5$ by using $13 = (1101)_2$, which means that the $(2, 5)$-request can be fulfilled by the path via subport 1 at stage 0, subport 1 at stage 1, subport 0 at stage 2, and subport 1 at stage 3. A routing algorithm is called *tag-based* if it uses a control tag to route a message. Most of the routing algorithms for MINs are tag-based, including those for GSENs. The routing algorithms proposed in this paper are also tag-based. Therefore, whenever a message is sent out, a control tag will be equipped with it.

Again, see Fig. 5. When a message is sent from i to j along P, the message enters a switch at stage $n-1-\ell$ via subport b_ℓ and leaves the switch via subport f_ℓ. On the other hand, when a message is sent from j to i along P, then the message enters a switch at stage $n-1-\ell$ via subport f_ℓ and leaves the switch via subport b_ℓ. The control tag

$$F = f_{n-1}2^{n-1} + f_{n-2}2^{n-2} + \cdots + f_0 2^0$$

is called a *forward control tag* for i to get to j. Most researchers simply called a forward control tag a control tag; here we add the word "forward" to specify that this control tag is used for sending a message in the forward direction, i.e., from the left-hand side of the GSEN to the right-hand side. Now let

$$B = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \cdots + b_0 2^0.$$

B is called a *backward control tag* and it is used for sending a message in the backward direction (from j to i). Clearly, $0 \le F < 2^n$ and $0 \le B < 2^n$.

Suppose F is given. In this paper, $P(i, F)$ denotes the path started from i and using the forward control tag F. Also, $B(i, F)$ denotes the backward control tag obtained from the path $P(i, F)$. Let

$$\mathcal{B}_F = \{B(i, F) \mid i = 0, 1, \ldots, N-1\}.$$

In the remaining discussion, $\oplus$ denotes the bitwise XOR operation. As a reference,

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0.$$

If $U = (u_{n-1}\, u_{n-2}\, \cdots\, u_0)_2$ and $V = (v_{n-1}\, v_{n-2}\, \cdots\, v_0)_2$, then we define

$$U \oplus V = (u_{n-1} \oplus v_{n-1}\ u_{n-2} \oplus v_{n-2}\ \cdots\ u_0 \oplus v_0)_2.$$

3. The proof of $N \le \mathcal{R}(N) \le \mathcal{R}_{sc}(N) = 2^n$

The purpose of this section is to prove that $N \le \mathcal{R}(N) \le \mathcal{R}_{sc}(N) = 2^n$. We first prove two lemmas.

Lemma 3.1. $N \le \mathcal{R}(N) \le \mathcal{R}_{sc}(N) \le 2^n$.

Proof. Given a network configuration, a permutation can be obtained. Thus a network configuration can be used to send N (personalized) messages simultaneously. The inequality $N \le \mathcal{R}(N)$ thus follows from that fact that N^2 messages have to be sent to fulfill all-to-all personalized exchange and each network configuration can send only N messages. The inequality $\mathcal{R}(N) \le \mathcal{R}_{sc}(N)$ is obvious. The inequality $\mathcal{R}_{sc}(N) \le 2^n$ follows from the fact that a GSEN has at most 2^n network configurations when the stage control technique is assumed. □

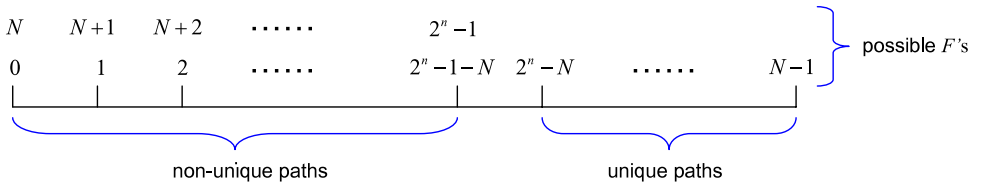

Fig. 6. Unique paths and multipaths.

In this paper, we call the process of transmitting all the messages to their next stage(s) a *round*. Thus in an n-stage MIN, it takes n rounds for a message to arrive its destination. In [13], Yang and Wang proved that the communication delay of all-to-all personalized exchange in a $(\log_2 N)$-stage MIN is $\Omega(N+\log_2 N)$. This is due to the fact that each of the N processors (say, processor j) has to receive N messages and it takes $\log_2 N$ rounds for the first message to arrive j and $N-1$ rounds for the remaining $N-1$ messages to arrive j. By similar arguments, we have the following lemma and its proof is omitted.

Lemma 3.2. *The communication delay of all-to-all communication in a GSEN is $\Omega(N+n)$, or equivalently, $\Omega(N+\log_2 N)$.*

Do notice that although $\Omega(N+n)=\Omega(N)$, we will still write $\Omega(N+n)$ instead of $\Omega(N)$ to emphasize that it takes n rounds for the first message to arrive its destination. In [5], Lan et al. considered GSENs with switches of size $d\times d$. By setting $d=2$, the following lemma can be obtained.

Lemma 3.3 (*[5]*). *Given i and F, the destination j of the path $P(i,F)$ is determined by*

$$j=(i\cdot 2^n+F) \mod N.$$

Moreover, the backward control tag B of the path $P(i,F)$ is given by

$$B=\left\lfloor \frac{i\cdot 2^n+F}{N} \right\rfloor.$$

When the stage control technique is assumed, the network configuration of a GSEN can be represented by a number as follows. Let c_ℓ denote the state, 0 for straight and 1 for cross, of all the switches at stage $n-1-\ell$. Then the network configuration of the GSEN can be represented by the number

$$C=c_{n-1}2^{n-1}+c_{n-2}2^{n-2}+\cdots+c_0 2^0$$

or by the binary number $(c_{n-1}\ c_{n-2}\ \cdots\ c_0)_2$. For example, the network configuration of the GSEN in Fig. 4(a) can be represented by 9 or by $(1001)_2$. Clearly, $0\le C<2^n$. Now we give the relation between F (a forward control tag), B (its corresponding backward control tag) and C (the network configuration).

Lemma 3.4. *When the stage control technique is assumed, F and B together uniquely determine the network configuration C and*

$$C=B\oplus F.$$

Proof. Consider stage $n-1-\ell$. Since the stage control technique is assumed, all switches in stage $n-1-\ell$ are of the same state. Let $C=c_{n-1}2^{n-1}+c_{n-2}2^{n-2}+\cdots+c_0 2^0$ be the network configuration and see Fig. 5. At stage $n-1-\ell$, a message enters subport b_ℓ and leaves subport f_ℓ. If $b_\ell=f_\ell$, then the state of the switch is straight; hence $c_\ell=0=b_\ell\oplus f_\ell$. If b_ℓ differs from f_ℓ (in this case, (b_ℓ,f_ℓ) is (0, 1) or (1, 0)), then the state of the switch is cross; hence $c_\ell=1=b_\ell\oplus f_\ell$. From the above, $C=B\oplus F$. □

We call a path a *unique path* if it is the unique path between its source and destination. The following lemma is important.

Lemma 3.5. *For all $0\le i<N$, path $P(i,F)$ is a unique path if and only if $2^n-N\le F<N$; in particular, $P(i,2^{n-1})$ and $P(i,2^{n-1}+1)$ are unique paths. (See Fig. 6 for illustration.)*

Proof. Let i and j be the source and destination of a message. Suppose there are two distinct paths $P(i,F_1), P(i,F_2)$ from i to j. Then, by Lemma 3.3, the difference between F_1 and F_2 is N. Without loss of generality, assume that $F_2-F_1=N$. Since $F_1\ge 0$, $F_2\ge N$ must hold. Since $F_2<2^n$, it follows that $F_1<2^n-N$. Thus $P(i,F)$ is a unique path if and only if $2^n-N\le F<N$. Since $2^n-N\le 2^{n-1}<N$, $P(i,2^{n-1})$ is a unique path. Since $2^n-N\le 2^{n-1}+1<N$, $P(i,2^{n-1}+1)$ is also a unique path. □

Lemma 3.6. $\mathcal{B}_{2^{n-1}}=\mathcal{B}_{2^{n-1}+1}$.

Proof. Let $0\le i<N$. Let $b_{n-1}f_{n-1}b_{n-2}f_{n-2}\cdots b_0 f_0$ be the sequence of subports passed by path $P(i,2^{n-1})$; see Fig. 5. Similarly, let $b'_{n-1}f'_{n-1}b'_{n-2}f'_{n-2}\cdots b'_0 f'_0$ be the sequence of subports passed by path $P(i,2^{n-1}+1)$. Since the binary representations of 2^{n-1} and $2^{n-1}+1$ differ only at their rightmost bits, $b_{n-1}f_{n-1}b_{n-2}f_{n-2}\cdots b_0 f_0$ and $b'_{n-1}f'_{n-1}b'_{n-2}f'_{n-2}\cdots b'_0 f'_0$ are identical except that $f_0\ne f'_0$. Hence $B(i,2^{n-1})=b_{n-1}b_{n-2}\cdots b_0=b'_{n-1}b'_{n-2}\cdots b'_0=B(i,2^{n-1}+1)$. Since $\mathcal{B}_{2^{n-1}}=\{B(i,2^{n-1})\mid i=0,1,\ldots,N-1\}$ and $\mathcal{B}_{2^{n-1}+1}=\{B(i,2^{n-1}+1)\mid i=0,1,\ldots,N-1\}$, we have this lemma. □

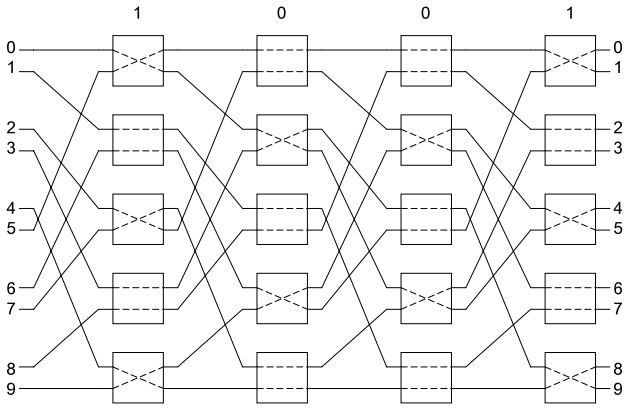

Fig. 7. Applying alternating stage control on a 10×10 GSEN; the shown network configuration is $A = 9 = (1001)_2$.

For convenience, if a number is in $\{0, 1, 2, \ldots, 2^n - 1\}$ but not in $\mathcal{B}_F$, then we call it a *hole* of $\mathcal{B}_F$. The following lemma shows that the elements of $\mathcal{B}_F$ are distributed *very uniformly* in $\{0, 1, 2, \ldots, 2^n - 1\}$.

Lemma 3.7. *For all $F \in \{0, 1, 2, \ldots, 2^n - 1\}$, $\mathcal{B}_F$ has no two consecutive holes.*

Proof. We will prove this lemma by showing that $B(0, F) \leq 1$, $B(N-1, F) \geq 2^n - 2$, and

$$B(i-1, F) + 1 \leq B(i, F) \leq B(i-1, F) + 2 \text{ for } i = 1, 2, \ldots, N-1.$$

Recall that $2^{n-1} < N \leq 2^n$ and $0 \leq F < 2^n$. By Lemma 3.3, $B(0, F) = \left\lfloor \frac{F}{N} \right\rfloor \leq 1$. Also, $B(N-1, F) = \left\lfloor \frac{(N-1)\cdot 2^n + F}{N} \right\rfloor \geq \left\lfloor \frac{(N-1)\cdot 2^n}{N} \right\rfloor \geq 2^n - 2$. Finally, consider i, where $1 \leq i \leq N-1$. By Lemma 3.3, $B(i-1, F) + 1 = \left\lfloor \frac{(i-1)\cdot 2^n + F}{N} \right\rfloor + 1 = \left\lfloor \frac{i\cdot 2^n + F}{N} - \frac{2^n}{N} \right\rfloor + 1 \leq \left\lfloor \frac{i\cdot 2^n + F}{N} \right\rfloor = B(i, F) = \left\lfloor \frac{(i-1)\cdot 2^n + F}{N} + \frac{2^n}{N} \right\rfloor \leq \left\lfloor \frac{(i-1)\cdot 2^n + F}{N} \right\rfloor + 2 = B(i-1, F) + 2$. □

Now we are ready to prove the main result of this section.

Theorem 3.8. $N \leq \mathcal{R}(N) \leq \mathcal{R}_{sc}(N) = 2^n$.

Proof. By Lemma 3.1, it suffices to prove that $\mathcal{R}_{sc}(N) \geq 2^n$. When the stage control technique is assumed, there are only 2^n possible network configurations: $0, 1, \ldots, 2^n - 1$. Thus to prove that $\mathcal{R}_{sc}(N) \geq 2^n$, it suffices to prove that each of the 2^n possible network configurations is required for every processor to receive N messages.

When the stage control technique is assumed, the network configuration C can be determined by an arbitrary path P set up by C. Moreover, if F and B are the forward and backward control tags used by P, then Lemma 3.4 tells us that $C = B \oplus F$. In the following, we will prove that for each C in $\{0, 1, \ldots, 2^n - 1\}$, at least one of the paths set up by C is a unique path and therefore C must be used in all-to-all personalized exchange. Suppose to the contrary there is a $\hat{C}$ in $\{0, 1, \ldots, 2^n - 1\}$ such that none of the paths set up by $\hat{C}$ is a unique path. Then consider $2^{n-1} \oplus \hat{C}$ and let $\hat{B} = 2^{n-1} \oplus \hat{C}$; consider $(2^{n-1} + 1) \oplus \hat{C}$ and let $\hat{B}' = (2^{n-1} + 1) \oplus \hat{C}$. We claim that $\hat{B} \notin \mathcal{B}_{2^{n-1}}$ and $\hat{B}' \notin \mathcal{B}_{2^{n-1}+1}$. Suppose this clam is not true. Then either $\hat{B} \in \mathcal{B}_{2^{n-1}}$ or $\hat{B}' \in \mathcal{B}_{2^{n-1}+1}$ or both. Suppose $\hat{B} \in \mathcal{B}_{2^{n-1}}$. Since $\hat{C} = \hat{B} \oplus 2^{n-1}$, by Lemma 3.5, $\hat{C}$ conducts a unique path, which contradicts with the assumption that none of the paths set up by $\hat{C}$ is a unique path. The case that $\hat{B}' \in \mathcal{B}_{2^{n-1}+1}$ can be proven similarly. Now we have the claim that $\hat{B} \notin \mathcal{B}_{2^{n-1}}$ and $\hat{B}' \notin \mathcal{B}_{2^{n-1}+1}$. By Lemma 3.6, $\mathcal{B}_{2^{n-1}} = \mathcal{B}_{2^{n-1}+1}$. Thus $\hat{B}' \notin \mathcal{B}_{2^{n-1}}$. Since $\hat{B}$ and $\hat{B}'$ differ by 1, they are two consecutive holes in $\mathcal{B}_{2^{n-1}}$; this contradicts with Lemma 3.7. Thus for each C in $\{0, 1, \ldots, 2^n - 1\}$, at least one of the paths set up by C is a unique path and therefore C must be used in all-to-all personalized exchange. So $\mathcal{R}_{sc}(N) \geq 2^n$. □

4. All-to-all personalized exchange of GSENs with $N \equiv 2 \pmod 4$

Throughout this section, unless other specified, subports 0 and 1 are the subports 0 and 1 on the right-hand side of a switch. We will propose an optimal all-to-all personalized exchange algorithm for GSENs with $N \equiv 2 \pmod 4$ and prove that $N = \mathcal{R}(N) < \mathcal{R}_{sc}(N) = 2^n$ if $N \equiv 2 \pmod 4$. We first introduce a variation of the stage control technique and we call it *alternating stage control*, meaning that the states of the switches of a stage *alternate between straight and cross*. See Fig. 7 for an illustration.

When alternating stage control is used, the network configuration of a GSEN can be represented by a number as follows. Let a_ℓ denote the states of the switches at stage $n-1-\ell$ such that

- $a_\ell = 0$ means the states are 0, 1, 0, 1, and so on;
- $a_\ell = 1$ means the states are 1, 0, 1, 0, and so on.

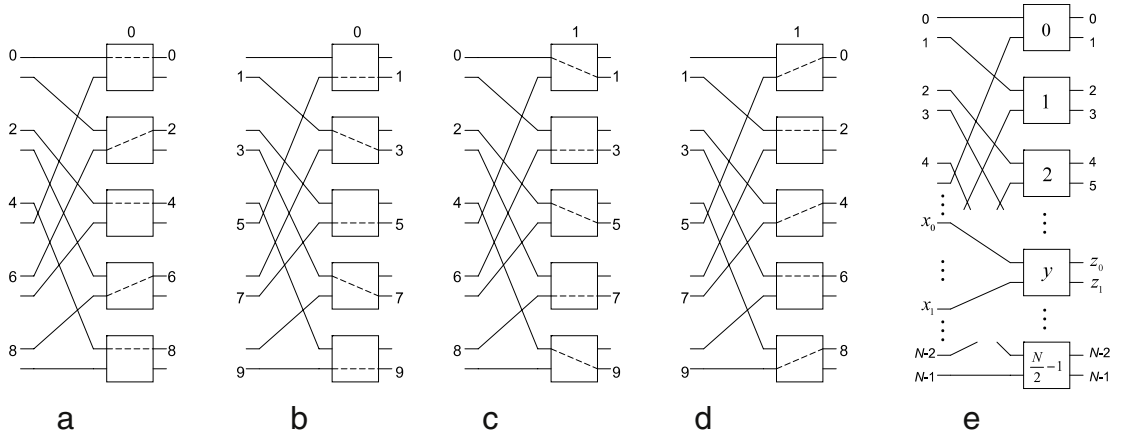

Fig. 8. (a) and (b): a stage in a 10×10 GSEN when $a_\ell = 0$. (c) and (d): a stage in a 10×10 GSEN when $a_\ell = 1$. (e) An illustration for the proof of Property $(*)$.

The network configuration of the GSEN can be represented by the number

$$A = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_0 2^0$$

or $(a_{n-1}\ a_{n-2}\ \ldots\ a_1\ a_0)_2$ in the binary form. Clearly, $0 \le A < 2^n$. We will call A an *alternating configuration*. When $N \equiv 2 \pmod 4$ and alternating stage control is used, the N input terminals and N output terminals of stage $n-1-\ell$ have the following property.

Property $(*)$ (*see Fig. 8 (a)–(d) for an illustration*).

1. *If $a_\ell = 0$, then $Even \xrightarrow{0} Even$, $Odd \xrightarrow{1} Odd$. That is, every even-numbered input terminal is connected to an even-numbered output terminal via subport 0, and every odd-numbered input terminal is connected to an odd-numbered output terminal via subport 1.*
2. *If $a_\ell = 1$, then $Even \xrightarrow{1} Odd$, $Odd \xrightarrow{0} Even$. That is, every even-numbered input terminal is connected to an odd-numbered output terminal via subport 1, and every odd-numbered input terminal is connected to an even-numbered output terminal via subport 0.*

Proof. Consider an arbitrary stage of a GSEN and an arbitrary switch y of this stage; see Fig. 8(e) for an illustration. Suppose input terminals x_0 and x_1 are connected to subports 0 and 1 of switch y, respectively. By the definition of a GSEN, $x_0 = y$ and $x_1 = y + \frac{N}{2}$ hold. Note that $\frac{N}{2}$ is an odd number since $N \equiv 2 \pmod 4$. Since $\frac{N}{2}$ is odd, one of x_0 and x_1 is even and the other one is odd. Now consider the output terminals z_0 and z_1 of switch y. Then z_0 is even and z_1 is odd.

Suppose $a_\ell = 0$ and y is even. Then x_0 is even (by the fact that $x_0 = y$) and x_0 is connected to z_0 (due to the setting of a_ℓ). Thus every even-numbered input terminal is connected to an even-numbered output terminal via subport 0. Now suppose $a_\ell = 0$ and y is odd. Then x_0 is odd (by the fact that $x_0 = y$) and x_0 is connected to z_1 (due to the setting of a_ℓ). Thus every odd-numbered input terminal is connected to an odd-numbered output terminal via subport 1. The case of $a_\ell = 1$ can be proven similarly. □

Do notice that Property $(*)$ holds only when $N \equiv 2 \pmod 4$ holds. Now we give other properties of alternating stage control.

Lemma 4.1. *Suppose $N \equiv 2 \pmod 4$ and alternating stage control is used. Then the following statements hold:*

1. *The forward control tags of even-numbered inputs are identical.*
2. *The forward control tags of odd-numbered inputs are identical.*

Proof. Let $A = (a_{n-1}\ a_{n-2}\ \cdots\ a_0)_2$ be the alternating configuration used. By Property $(*)$, all the messages sent out from inputs 0, 2, 4, …, $N-2$ are via subports 0 of switches at stage $n-1-\ell$ if $a_\ell = 0$ and via subports 1 of switches at stage $n-1-\ell$ if $a_\ell = 1$. Thus statement 1 holds. By Property $(*)$, all the messages sent out from inputs 1, 3, 5, …, $N-1$ are via subports 1 of switches at stage $n-1-\ell$ if $a_\ell = 0$ and via subports 0 of switches at stage $n-1-\ell$ if $a_\ell = 1$. Thus we have statement 2. □

Theorem 4.2. *Suppose $N \equiv 2 \pmod 4$ and alternating stage control is used. Let A be a given alternating configuration, F be the forward control tag of any even-numbered input, and $\overline{F}$ be the forward control tag of any odd-numbered input. Then:*

(i) $F \oplus \overline{F} = (11\cdots1)_2$;
(ii) $A = F \oplus \left\lfloor \frac{F}{2} \right\rfloor$;
(iii) $F = A \oplus \left\lfloor \frac{A}{2} \right\rfloor \oplus \left\lfloor \frac{A}{2^2} \right\rfloor \oplus \cdots \oplus \left\lfloor \frac{A}{2^{n-1}} \right\rfloor$.

Proof. First consider (i). Let $F = (f_{n-1}\, f_{n-2}\, \cdots\, f_0)_2$ and $A = (a_{n-1}\, a_{n-2}\, \cdots\, a_0)_2$. By Property (∗), if messages from even-numbered inputs are via subport f_ℓ at stage $n-1-\ell$, then messages from inputs odd are via subport $1-f_\ell$ at stage $n-1-\ell$, ($\ell = n-1, n-2, \ldots, 0$). Thus $F \oplus \overline{F} = (11\cdots1)_2$. Now consider (ii). Clearly, $a_{n-1} = f_{n-1}$. For $\ell = n-2, n-3, \ldots, 0$, by Property (∗), we have:

- If $a_\ell = 0$, then $f_\ell = 0$ whenever $f_{\ell+1} = 0$ and $f_\ell = 1$ whenever $f_{\ell+1} = 1$.
- If $a_\ell = 1$, then $f_\ell = 0$ whenever $f_{\ell+1} = 1$ and $f_\ell = 1$ whenever $f_{\ell+1} = 0$.

Thus $a_\ell = f_\ell \oplus f_{\ell+1}$ for $\ell = n-2, n-3, \ldots, 0$. Therefore

$$\begin{aligned} A = (a_{n-1}\, a_{n-2}\, \ldots\, a_1\, a_0)_2 &= (f_{n-1}\ f_{n-2}\oplus f_{n-1}\ f_{n-3}\oplus f_{n-2}\ \ldots\ f_0\oplus f_1)_2 \\ &= (f_{n-1}\oplus 0\ f_{n-2}\oplus f_{n-1}\ f_{n-3}\oplus f_{n-2}\ \ldots\ f_0\oplus f_1)_2 \\ &= (f_{n-1}\ f_{n-2}\ f_{n-3}\ \ldots\ f_0)_2 \oplus (0\ f_{n-1}\ f_{n-2}\ \ldots\ f_1)_2 = F \oplus \left\lfloor \frac{F}{2} \right\rfloor. \end{aligned}$$

Finally, consider (iii). Then $f_\ell = a_\ell \oplus a_{\ell+1} \oplus \cdots \oplus a_{n-1}$ for $\ell = n-2, n-3, \ldots, 0$. Thus $F = A \oplus \lfloor \frac{A}{2} \rfloor \oplus \lfloor \frac{A}{2^2} \rfloor \oplus \cdots \oplus \lfloor \frac{A}{2^{n-1}} \rfloor$. □

Theorem 4.2(ii) gives a one-to-one correspondence between A and F; for convenience, let A_F denote the alternating configuration corresponding to F. When $F = k$,

$$A_k = k \oplus \left\lfloor \frac{k}{2} \right\rfloor.$$

Lemma 4.3. *If $N \equiv 2 \pmod 4$ and the given GSEN is set by alternating configuration A_k, then the forward control tags of even-numbered inputs are k and the forward control tags of odd-numbered inputs are $2^n - 1 - k$.*

Proof. This lemma follows from Lemma 4.1, $A_k = k \oplus \lfloor \frac{k}{2} \rfloor$, and Theorem 4.2(i). □

Now we prove a theorem, which is the foundation of our algorithms.

Theorem 4.4. *Suppose $N \equiv 2 \pmod 4$. Then the N alternating configurations $A_0, A_1, \ldots, A_{N-1}$ ensure that every input i can get to every output j; in other words, $A_0, A_1, \ldots, A_{N-1}$ can fulfill an all-to-all communication in a GSEN.*

Proof. Let i be an arbitrary input. For $k = 0, 1, \ldots, N-1$, let j_k be the destination of i when the network configuration is set according to A_k. First consider the case that i is even. By Lemmas 3.3 and 4.3, $j_k = (i \cdot 2^n + k) \mod N$. Since $A_0, A_1, \ldots, A_{N-1}$ ensure that k varies from 0 to $N-1$ and $j_k = (i \cdot 2^n + k) \mod N$, it follows that i can get to every output. Now consider the case that i is odd. By Lemmas 3.3 and 4.3, $j_k = (i \cdot 2^n + 2^n - 1 - k) \mod N$. Since $A_0, A_1, \ldots, A_{N-1}$ ensure that k varies from 0 to $N-1$ and $j_k = (i \cdot 2^n + 2^n - 1 - k) \mod N$, it follows that i can get to every output. □

For example, the 10 alternating configurations $A_0 = 0, A_1 = 1, A_2 = 3, A_3 = 2, A_4 = 6, A_5 = 7, A_6 = 5, A_7 = 4, A_8 = 12, A_9 = 13$ can fulfill an all-to-all communication in a 10×10 GSEN. Note that $A_0, A_1, \ldots, A_{N-1}$ are not the only way to fulfill an all-to-all communication in a GSEN. In fact, any N consecutive integers in $0, 1, \ldots, 2^n - 1$ can fulfill an all-to-all communication.

The purpose of this paper is to propose an optimal all-to-all personalized exchange algorithm for GSENs. However, since there is no all-to-all broadcast algorithm for GSENs, we will also propose one. Therefore, in the following, three algorithms will be proposed. The first algorithm fulfills all-to-all broadcast in GSENs. The second algorithm gives a preprocessing of the third algorithm. And the third algorithm fulfills all-to-all personalized exchange in GSENs.

Algorithm 1 : an algorithm to fulfill all-to-all broadcast in a GSEN with $N \equiv 2 \pmod 4$

```
1: for each processor i (0 ≤ i < N) do in parallel
2:   Processor i prepares a broadcast message;
3:   for k = 0 to N − 1 do in sequential
4:     Equip the broadcast message of processor i with the forward control tag k if i is even and 2^n − 1 − k if i is odd;
5:     Transmit the message;
6:   endfor
7: endfor
```

The correctness of Algorithm 1 follows from Lemma 4.3 and Theorem 4.4. The communication delay of Algorithm 1 is $O(N + n)$ since each of the N processors can receive its first message in n rounds and receive the remaining $N-1$ messages in $N-1$ rounds. By Lemma 3.2, Algorithm 1 is optimal.

All-to-all personalized exchange is much more complicated than all-to-all broadcast. In all-to-all personalized exchange, a source has to prepare a personalized message for each of its N destinations. Therefore, before a message is sent out, the source of the message has to know which output will be its current destination so that a personalized message can be prepared. Algorithm 2 is designed to overcome this difficulty. This algorithm constructs a matrix called *destination matrix* $D = (d_{i,k})$ so that $d_{i,k} = j$ if and only if the message sent out from processor i at round k arrives processor j.

The following theorem proves the correctness and gives the time complexity of Algorithm 2.

Algorithm 2 : an algorithm to construct the destination matrix $D = (d_{i,k})$ for a GSEN with $N \equiv 2 \pmod 4$

```
1: n ← ⌈log2 N⌉;
2: power ← 2^n;
3: for each processor i (0 ≤ i < N) do in sequential
4:   if i is even then m ← (i · power) mod N; else m ← ((i + 1) · power − 1) mod N; endif
5:   for k = 0 to N − 1 do in sequential
6:     if i is even then d_{i,k} ← (m + k) mod N; else d_{i,k} ← (m − k) mod N; endif
7:   endfor
8: endfor
```

Theorem 4.5. *Algorithm 2 constructs a matrix $D = (d_{i,k})$ so that $d_{i,k} = j$ if and only if the message sent out from processor i at round k arrives processor j. Moreover, it takes $O(N^2)$ time.*

Proof. To prove the correctness of Algorithm 2, it suffices to show that the message sent out from processor i at round k (see Algorithm 3 for round k) arrives processor $(m + k) \bmod N$ if i is even and arrive processor $(m - k) \bmod N$ if i is odd. Note that in Algorithm 3, we will use $A_0, A_1, \ldots, A_{N-1}$ to fulfill all-to-all personalized exchange. By Lemma 4.3, A_k contributes an even-numbered processor i the forward control tag k and it contributes an odd-numbered processor i the forward control tag $2^n - 1 - k$. Suppose i is even. Then at round k, the message sent out from processor i will be equipped with the forward control tag k; by Lemma 3.3, the destination is

$$j = (i \cdot 2^n + k) \bmod N = (i \cdot \text{power} + k) \bmod N = (m + k) \bmod N.$$

Now suppose i is odd. By Lemma 4.3, the message sent out from processor i will be equipped with the forward control tag $2^n - 1 - k$; by Lemma 3.3, the destination is

$$j = (i \cdot 2^n + 2^n - 1 - k) \bmod N = ((i + 1) \cdot \text{power} - 1 - k) \bmod N = (m - k) \bmod N.$$

It is not difficult to see that Algorithm 2 takes $O(N^2)$ time. We have this theorem. □

Consider the GSEN in Fig. 2 for an example of Algorithm 2. Then the matrix D constructed is:

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 0 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 \\ 3 & 2 & 1 & 0 & 9 & 8 & 7 & 6 & 5 & 4 \\ 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 \\ 5 & 4 & 3 & 2 & 1 & 0 & 9 & 8 & 7 & 6 \\ 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 9 & 8 \\ 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}.$$

Note that the matrix D needs to be constructed only once and therefore can be viewed as one of the system parameters. Thus the time complexity of Algorithm 2 is *not* included in the communication delay. Now we are ready to propose our all-to-all personalized exchange algorithm; see Algorithm 3.

Algorithm 3 : an algorithm to fulfill all-to-all personalized exchange in a GSEN with $N \equiv 2 \pmod 4$

```
1: for each processor i (0 ≤ i < N) do in parallel
2:   for k = 0 to N − 1 do in sequential //comment: round k
3:     Processor i prepares a personalized message for processor d_{i,k};
4:     Equip the personalized message with the forward control tag k if i is even and 2^n − 1 − k if i is odd;
5:     Transmit the message;
6:   endfor
7: endfor
```

The following theorem proves the correctness and gives the time complexity of Algorithm 3.

Theorem 4.6. *Algorithm 3 fulfills all-to-all personalized exchange in a GSEN with $N \equiv 2 \pmod 4$. Moreover, it takes $O(N + n)$ time.*

Proof. Algorithm 3 prepares a personalized message according to the matrix D, which is constructed by Algorithm 2. Thus, by Theorems 4.4 and 4.5, Algorithm 3 fulfills all-to-all personalized exchange for GSENs with $N \equiv 2 \pmod 4$. This algorithm takes $O(N + n)$ time since each of the N processors can receive its first personalized message in n rounds and receive the remaining $N - 1$ personalized messages in $N - 1$ rounds. □

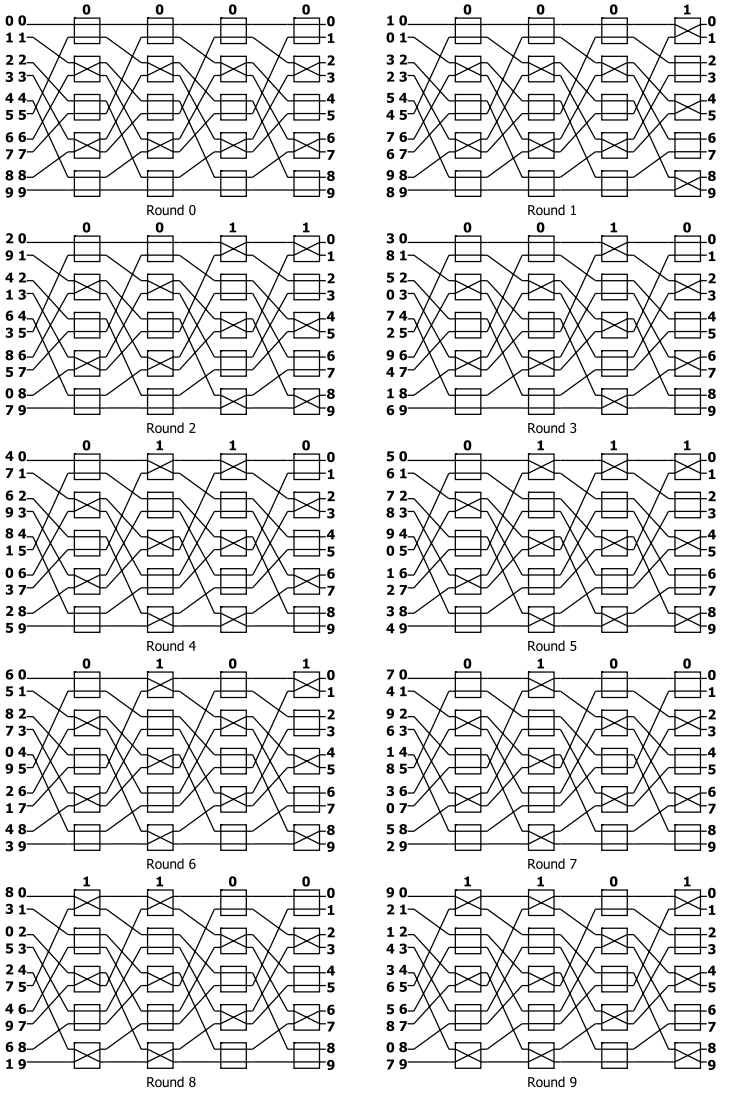

Fig. 9. An example of Algorithm 3.

By Lemma 3.2 and Theorem 4.6, we have the following corollary.

Corollary 4.7. *Algorithm 3 is optimal.*

Fig. 9 shows how Algorithm 3 fulfills all-to-all personalized exchange for the GSEN in Fig. 2. Take round 2 in Fig. 9 for an example. The 0-1 bits 0011 above stages 0, 1, 2, 3 denote the alternating configuration for round 2, which is $(0011)_2 = 3 = A_2$. The numbers on the left-hand side denote the destinations of personalized messages. Thus, at round 2, processor 0 sends a personalized message to processor 2, processor 1 sends a personalized message to processor 9, processor 2 sends a personalized message to processor 4, . . . , and processor 9 sends a personalized message to processor 7. Recall that $A_0 = 0, A_1 = 1, A_2 = 3, A_3 = 2, A_4 = 6, A_5 = 7, A_6 = 5, A_7 = 4, A_8 = 12, A_9 = 13$ can fulfill an all-to-all communication in a 10×10 GSEN. The 10 alternating configurations and the destinations of the messages are shown on the left-hand side of the GSEN for rounds 0, 1, . . . , 9 in Fig. 9.

Note that it is possible to combine Algorithms 2 and 3 and to avoid the construction of matrix D. See Algorithm 4 below. Now we end this section by proving the following theorem.

Theorem 4.8. $N = \mathcal{R}(N) < \mathcal{R}_{sc}(N) = 2^n$ if $N \equiv 2 \pmod 4$.

Proof. Since Algorithm 3 can fulfill all-to-all personalized exchange by using N network configurations, namely, $A_0, A_1, \ldots, A_{N-1}$, we have $\mathcal{R}(N) \leq N$. By Theorem 3.8 and by the fact that $\mathcal{R}(N) \leq N$ for $N \equiv 2 \pmod 4$, we have this theorem. □

Algorithm 4 : yet another algorithm to fulfill all-to-all personalized exchange in a GSEN with $N \equiv 2 \pmod 4$

```
1: for each processor i (0 ≤ i < N) do in parallel
2:   n ← ⌈log2 N⌉;
3:   power ← 2^n;
4:   if i is even then m ← (i · power) mod N; else m ← ((i + 1) · power − 1) mod N; endif
5:   for k = 0 to N − 1 do in sequential //comment: round k
6:     if i is even then j ← (m + k) mod N; else j ← (m − k) mod N; endif
7:     Processor i prepares a personalized message for processor j;
8:     Equip the personalized message with the forward control tag k if i is even and 2^n − 1 − k if i is odd;
9:     Transmit the message;
10:  endfor
11: endfor
```

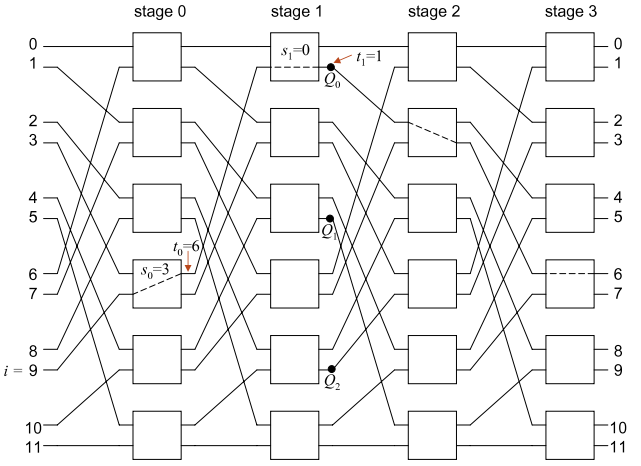

Fig. 10. A 12 × 12 GSEN, switches s_0 and s_1, terminals t_0 and t_1, and $\mathcal{Q} = \{Q_0, Q_1, Q_2\}$.

5. The value of $\mathcal{R}(N)$ when $N \equiv 0 \pmod 4$

The purpose of this section is to obtain $\mathcal{R}(N)$ for all $N \equiv 0 \pmod 4$. Recall that each stage of a GSEN consists of the perfect shuffle on N terminals followed by $N/2$ switches, the N terminals are numbered $0, 1, \ldots, N-1$, and the perfect shuffle operation on the N terminals is the permutation π defined by $\pi(i) = (2 \cdot i + \lfloor \frac{2 \cdot i}{N} \rfloor) \bmod N, \quad 0 \le i < N$. We first have a lemma.

Lemma 5.1. *Suppose $k \ge 2$, $N \equiv 0 \pmod{2^k}$, and $N \not\equiv 0 \pmod{2^{k+1}}$. Let i be an arbitrary input of a given $N \times N$ GSEN. If the forward control tag $F = f_{n-1}2^{n-1} + f_{n-2}2^{n-2} + \cdots + f_0 2^0$ used by i starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$), then the terminal reached by i immediately after stage $k-1$ is $(i2^k + 2^{k-1} - 1) \bmod N$.*

Proof. Each stage of a GSEN has $N/2$ switches; we suppose these $N/2$ switches are labeled $0, 1, \ldots, N/2-1$. Consider the path $P(i, F)$ and the switches and terminals on the path. Let s_ℓ be the label of the switch at stage ℓ reached by $P(i, F)$. Let t_ℓ be the terminal immediately after stages ℓ that is reached by $P(i, F)$. See Fig. 10 for an illustration of the $N = 12$ and $k = 2$ case. By the perfect shuffle operation, $s_0 = i \bmod N/2$. Since $f_{n-1} = 0$, we have $t_0 = 2s_0 = (2i) \bmod N$. Again, by the perfect shuffle operation, $s_1 = t_0 \bmod N/2 = (2i) \bmod N/2$. Since $f_{n-2} = 1$, we have $t_1 = 2s_1 + 1 = (4i + 1) \bmod N = (i2^2 + 2^1 - 1) \bmod N$. In general, we assume $\ell \ge 1$. Then we have $s_\ell = t_{\ell-1} \bmod N/2$ and $t_\ell = 2s_\ell + f_{n-1-\ell}$. Continuing in this way, we have

$$t_{k-1} = (i2^k + f_{n-1}2^{k-1} + \cdots + f_{n-k}2^0) \bmod N = (i2^k + 2^{k-2} + \cdots + 2^0) \bmod N = (i2^k + 2^{k-1} - 1) \bmod N.$$

Hence this lemma holds. □

For $r = 0, 1, \ldots, \frac{N}{2^k} - 1$, let Q_r denote the terminal $r2^k + 2^{k-1} - 1$ immediately after stage $k-1$ and see Fig. 10 for an illustration of the $N = 12$ and $k = 2$ case. Let $\mathcal{Q} = \{Q_r \mid r = 0, 1, \ldots, \frac{N}{2^k} - 1\}$. We say a routing path *passes through* $\mathcal{Q}$ if it passes through one of the terminals in $\mathcal{Q}$.

Lemma 5.2. *Suppose $k \ge 2$, $N \equiv 0 \pmod{2^k}$, $N \not\equiv 0 \pmod{2^{k+1}}$, and consider an $N \times N$ GSEN. A routing path passes through $\mathcal{Q}$ if and only if the forward control tag $F = f_{n-1}2^{n-1} + f_{n-2}2^{n-2} + \cdots + f_0 2^0$ used by this path starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$).*

Proof. Assume that the given routing path is from input i. Then this routing path is the path $P(i, F)$.
(Necessity) First suppose $P(i, F)$ passes through the terminal Q_r in $\mathcal{Q}$. Then by the perfect shuffle operation, we have

$$(i2^k + f_{n-1}2^{k-1} + \cdots + f_{n-k}) \bmod N = r2^k + 2^{k-1} - 1.$$

Since $2^k | N$, we can take modulo 2^k for both sides of the above equation and obtain

$$(i2^k + f_{n-1}2^{k-1} + \cdots + f_{n-k} \bmod N) \bmod 2^k = (r2^k + 2^{k-1} - 1) \bmod 2^k = 2^{k-1} - 1,$$

which implies $f_{n-1}2^{k-1} + f_{n-2}2^{k-2} + \cdots + f_{n-k} = 2^{k-1} - 1$, i.e., $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$).
(Sufficiency) Suppose the forward control tag F starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$). Then by Lemma 5.1, the terminal reached by i immediately after stage $k-1$ will be $(i2^k + 2^{k-1} - 1) \bmod N$, which is $Q_{i \bmod \frac{N}{2^k}}$. Therefore $P(i, F)$ passes through $\mathcal{Q}$. □

Recall that $2^{n-1} < N \leq 2^n$. The following lemma requires N to satisfy $2^{n-1} + 2^{n-k} \leq N \leq 2^n$.

Lemma 5.3. *Suppose $k \geq 2$, $N \equiv 0 \pmod{2^k}$, $N \not\equiv 0 \pmod{2^{k+1}}$, and consider an $N \times N$ GSEN. If $2^{n-1} + 2^{n-k} \leq N \leq 2^n$ and the forward control tag $F = f_{n-1}2^{n-1} + f_{n-2}2^{n-2} + \cdots + f_0 2^0$ used by a path starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$), then this path is a unique path.*

Proof. Note that if F starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$), then $2^{n-1} - 2^{n-k} \leq F < 2^{n-1}$. Assume that the given routing path is from input i. Then this routing path is the path $P(i, F)$. By Lemma 3.5, $P(i, F)$ is a unique path if and only if $2^n - N \leq F < N$. Since

$$2^n - N \leq 2^n - 2^{n-1} - 2^{n-k} = 2^{n-1} - 2^{n-k} \leq F < 2^{n-1} < N,$$

$P(i, F)$ is a unique path for each $2^{n-1} - 2^{n-k} \leq F < 2^{n-1}$. Hence this lemma holds. □

Now we are ready to propose our result for $\mathcal{R}(N)$ with $N \equiv 0 \pmod 4$.

Theorem 5.4. *$\mathcal{R}(N) = \mathcal{R}_{sc}(N) = 2^n$ if $k \geq 2$, $N \equiv 0 \pmod{2^k}$, $N \not\equiv 0 \pmod{2^{k+1}}$, and $2^{n-1} + 2^{n-k} \leq N \leq 2^n$.*

Proof. Assume $k \geq 2$, $N \equiv 0 \pmod{2^k}$, $N \not\equiv 0 \pmod{2^{k+1}}$, and $2^{n-1} + 2^{n-k} \leq N \leq 2^n$. By Theorem 3.8, it suffices to prove that $\mathcal{R}(N) \geq 2^n$. In any all-to-all communication of a GSEN, a total of N^2 routing paths have to be established. Let i be an arbitrary input and let $F = f_{n-1}2^{n-1} + f_{n-2}2^{n-2} + \cdots + f_0 2^0$ be an arbitrary forward control tag such that F starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$). Since F starts with $f_{n-1} = 0$ and $f_{n-t} = 1$ (for $t = 2, 3, \ldots, k$), we have $2^{n-1} - 2^{n-k} \leq F < 2^{n-1}$ and there are a total of 2^{n-k} such F's. By Lemma 5.3, $P(i, F)$ is a unique path. Since $2^{n-1} - 2^{n-k} \leq F < 2^{n-1}$, the number of such unique paths $P(i, F)$ is $N \cdot 2^{n-k}$. Let $\mathcal{U}$ denote the set of these $N \cdot 2^{n-k}$ unique paths. Then, in any all-to-all communication, all of the paths in $\mathcal{U}$ must appear. By Lemma 5.2, all of the paths in $\mathcal{U}$ will pass through $\mathcal{Q}$. Recall that given a network configuration, a permutation between the inputs and outputs can be obtained. Therefore, given a network configuration, N routing paths can be established. By Lemma 5.1, any network configuration can establish only $N/2^k$ routing paths in $\mathcal{U}$. Therefore $\mathcal{R}(N) \geq \frac{N \cdot 2^{n-k}}{N/2^k} = 2^n$. □

By Theorem 5.4, $\mathcal{R}(12) = 16$, $\mathcal{R}(24) = 32$, $\mathcal{R}(28) = 32$, $\mathcal{R}(40) = 64$, $\mathcal{R}(80) = 128$, and $\mathcal{R}(144) = 256$. The first $\mathcal{R}(N)$ that cannot be determined by Theorems 4.8 and 5.4 is $\mathcal{R}(20)$; we will determine it after introducing a variation of the alternating stage control technique; we call it *doubly alternating stage control*, meaning that the states of the switches of a stage *alternate between two straight states and two cross states*. The network configuration obtained by doubly alternating stage control is called a *doubly alternating configuration* and it can be represented by the number

$$A' = a'_{n-1}2^{n-1} + a'_{n-2}2^{n-2} + \cdots + a'_0 2^0$$

as follows. Let a'_ℓ denote the states of the switches at stage $n-1-\ell$ such that

- $a'_\ell = 0$ means the states are 0, 0, 1, 1, 0, 0, 1, 1, and so on.
- $a'_\ell = 1$ means the states are 1, 1, 0, 0, 1, 1, 0, 0, and so on.

Obviously, $0 \leq A' < 2^n$. Now we are ready to determine $\mathcal{R}(20)$.

Theorem 5.5. *$\mathcal{R}(20) = 24$.*

Proof. We first prove that $\mathcal{R}(20) \geq 24$. In any all-to-all communication of a 20×20 GSEN, a total of $20^2 = 400$ routing paths have to be established. To prove $\mathcal{R}(20) \geq 24$, we claim that 400 routing paths are not sufficient to fulfill an all-to-all communication in a 20×20 GSEN and at least $400 + 80 = 480$ routing paths have to be established in order to fulfill an all-to-all communication. If this claim is true, then since a network configuration can establish only 20 routing paths, we have $\mathcal{R}(20) \geq \frac{480}{20} = 24$. Now we prove this claim.

Let i be an arbitrary input and let $F = f_{n-1}2^{n-1} + f_{n-2}2^{n-2} + \cdots + f_0 2^0$ be an arbitrary forward control tag. By Lemma 3.5, $P(i, F)$ is a unique path if and only if $12 \leq F \leq 19$. Hence each input i contributes 8 unique paths $P(i, 12)$, $P(i, 13), \ldots, P(i, 19)$. Thus there are a total of 160 unique paths; we illustrate all of these 160 unique paths in Fig. 11. In this proof, states of switches at stage 2 play an important role. Denote the 10 switches at stage 2 by $S_0, S_1, \ldots, S_9$. Now we

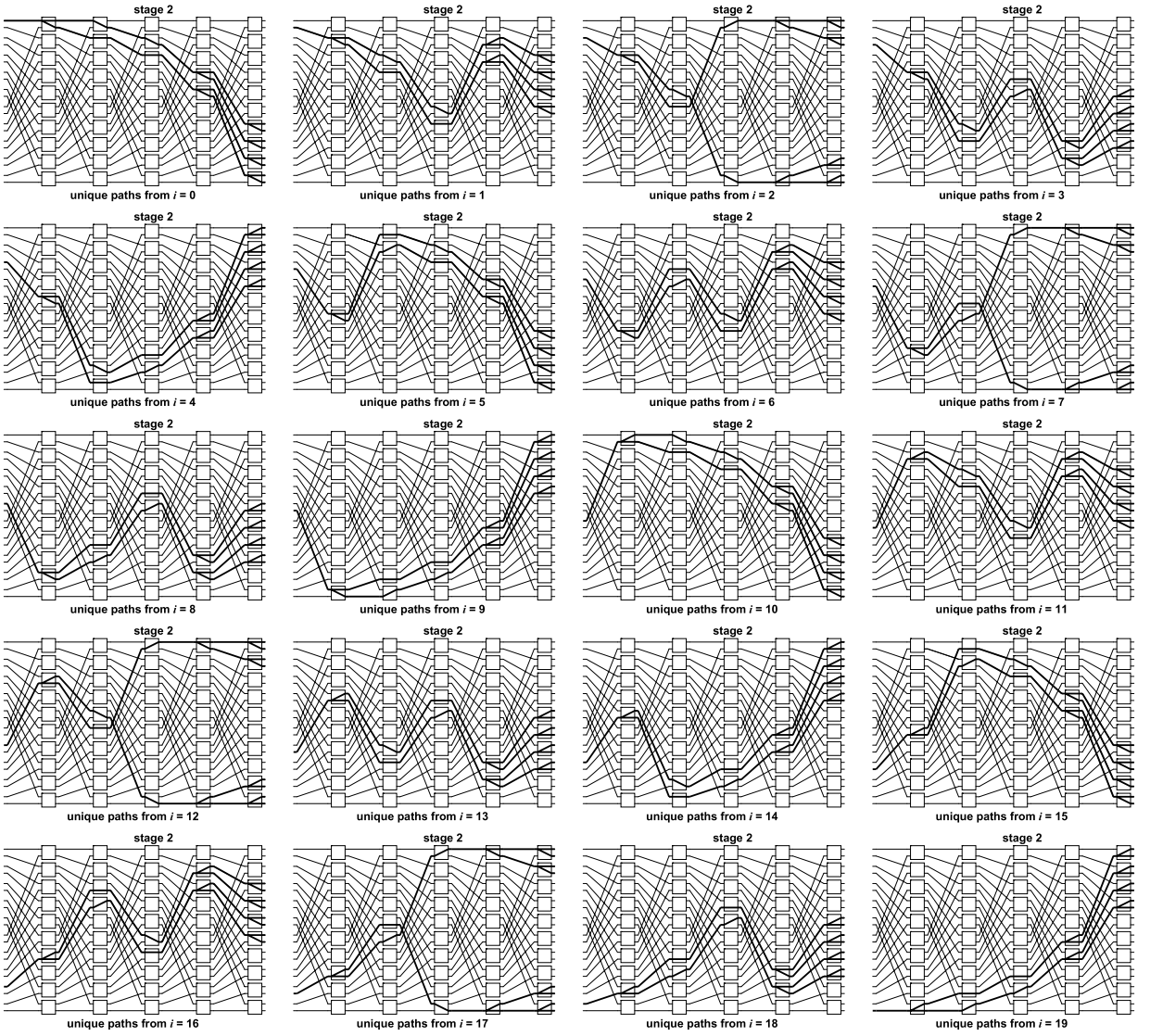

Fig. 11. The 160 unique paths in a 20 × 20 GSEN; each input i contributes 8 unique paths.

define types 00, 01, 10, and 11, according to the connection inside a switch at stage 2 as follows. A path is said to be *of type* xy, where x,y ∈ {0,1}, if the connection inside the switch (passed by the path) at stage 2 is from subport x to subport y. The following two facts can be observed from Fig. 11.

Fact 1: All of the unique paths passing through S_0, S_4, and S_8 are of type 10, through S_1, S_5, and S_9 are of type 01, through S_2 and S_6 are of type 00, and through S_3 and S_7 are of type 11. (See Fig. 12(a).)

Fact 2: Each switch at stage 2 has exactly 16 unique paths passing through it. More precisely, let $\mathcal{U}_i$ denote the set of all 16 unique paths passing through S_i. Then

$$\mathcal{U}_0 = \{P(i,F) \mid i = 2, 7, 12, 17 \text{ and } F = 16, 17, 18, 19\},\ \mathcal{U}_1 = \{P(i,F) \mid i = 0, 5, 10, 15 \text{ and } F = 12, 13, 14, 15\},$$
$$\mathcal{U}_2 = \{P(i,F) \mid i = 0, 5, 10, 15 \text{ and } F = 16, 17, 18, 19\},\ \mathcal{U}_3 = \{P(i,F) \mid i = 3, 8, 13, 18 \text{ and } F = 12, 13, 14, 15\},$$
$$\mathcal{U}_4 = \{P(i,F) \mid i = 3, 8, 13, 18 \text{ and } F = 16, 17, 18, 19\},\ \mathcal{U}_5 = \{P(i,F) \mid i = 1, 6, 11, 16 \text{ and } F = 12, 13, 14, 15\},$$
$$\mathcal{U}_6 = \{P(i,F) \mid i = 1, 6, 11, 16 \text{ and } F = 16, 17, 18, 19\},\ \mathcal{U}_7 = \{P(i,F) \mid i = 4, 9, 14, 19 \text{ and } F = 12, 13, 14, 15\},$$
$$\mathcal{U}_8 = \{P(i,F) \mid i = 4, 9, 14, 19 \text{ and } F = 16, 17, 18, 19\},\ \mathcal{U}_9 = \{P(i,F) \mid i = 2, 7, 12, 17 \text{ and } F = 12, 13, 14, 15\}.$$

By Fact 1, in a network configuration, switch S_0 has to be set to cross to let a unique path in $\mathcal{U}_0$ passing through it. Let $\mathcal{N}_0$ denote the set of paths of passing through S_0 which are of type 01; see Fig. 12(b). Also by Fact 1, in a network configuration, switch S_3 has to be set to straight to let a unique path in $\mathcal{U}_3$ passing through it. Let $\mathcal{N}_3$ denote the set of paths passing through S_3 which are of type 00; see Fig. 12(c). Let $I \times J$*-requests* denote the set of all (i, j)-requests with $i \in I$ and $j \in J$. It

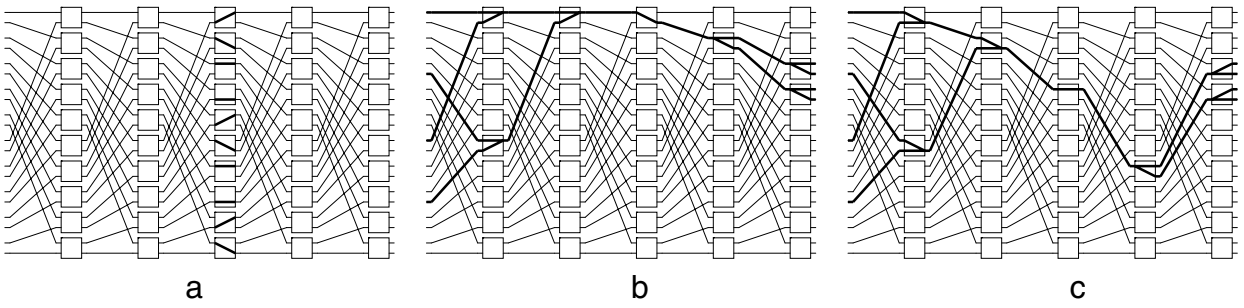

Fig. 12. (a) The setting of switches at stage 2 when unique paths pass through them. (b) The set of paths in $\mathcal{N}_0$. These paths fulfill $\{0, 5, 10, 15\} \times \{4, 5, 6, 7\}$-requests. (c) The set of paths in $\mathcal{N}_3$. These paths also fulfill $\{0, 5, 10, 15\} \times \{4, 5, 6, 7\}$-requests.

can be observed from Fig. 12(b)(c) that $\mathcal{N}_0$ and $\mathcal{N}_3$ both fulfill $\{0, 5, 10, 15\} \times \{4, 5, 6, 7\}$-requests. Thus when the 32 unique paths in $\mathcal{U}_0 \cup \mathcal{U}_3$ are fulfilled, the 32 paths in $\mathcal{N}_0 \cup \mathcal{N}_3$ are also established; however, $\mathcal{N}_0 \cup \mathcal{N}_3$ fulfills at most 16 routing requests and at least 16 routing requests are repeated. The same situation also occurs when the 32 unique paths in $\mathcal{U}_1 \cup \mathcal{U}_8$, in $\mathcal{U}_2 \cup \mathcal{U}_5$, in $\mathcal{U}_4 \cup \mathcal{U}_7$, and in $\mathcal{U}_6 \cup \mathcal{U}_9$ are established. From the above, a total of $16 \cdot 5 = 80$ routing requests are repeated. Hence to fulfill an all-to-all communication in a 20×20 GSEN, at least $400 + 80 = 480$ routing requests have to be fulfilled, i.e., 480 routing paths have to be established.

Now we prove $\mathcal{R}(20) \leq 24$ by showing that an all-to-all communication in a 20×20 GSEN can be fulfilled in 24 network configuration. A 20×20 GSEN has 32 doubly alternating configurations. Consider these 32 doubly alternating configurations. It is not difficult to check that $A' = 0$ and $A' = 17$ obtain the same permutation and hence only one of them is needed in an all-to-all communication. Each of the following pairs of doubly alternating configurations also obtain the same permutation and hence only one in each pair is needed in an all-to-all communication: $A' = 1$ and $A' = 16$, $A' = 2$ and $A' = 19$, $A' = 3$ and $A' = 18$, $A' = 8$ and $A' = 25$, $A' = 9$ and $A' = 24$, $A' = 10$ and $A' = 27$, and $A' = 11$ and $A' = 26$. By removing one doubly alternating configuration from each of the above eight pairs, we have a set $\mathcal{A}'$ containing 24 doubly alternating configurations; in particular, we can choose $\mathcal{A}' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31\}$. In Fig. 13, we show all the permutations obtained by applying the network configurations in $\mathcal{A}'$. It is not difficult to see that $\mathcal{A}'$ fulfills an all-to-all communication in a 20×20 GSEN. Hence $\mathcal{R}(20) \leq 24$. □

6. Concluding remarks

The shuffle-exchange network has been proposed as a popular architecture for MINs. The generalized shuffle-exchange networks (GSEN) is a generalization of the shuffle-exchange network. We follow the convention used in [1,2,10] that an $N \times N$ GSEN has exactly $\lceil \log_2 N \rceil$ stages. Based on this convention, we define $n = \lceil \log_2 N \rceil$ and we have $2^{n-1} < N \leq 2^n$.

In this paper we consider the all-to-all personalized exchange problem in GSENs. Since a GSEN does not have the unique path property, previous algorithms [9,13] cannot be used. To our knowledge, no one has studied all-to-all personalized exchange in MINs which do not have the unique path property and do not satisfy $N = 2^n$. An optimal algorithm and several bounds on $\mathcal{R}(N)$ and $\mathcal{R}_{sc}(N)$ have been proposed in this paper; recall that $\mathcal{R}(N)$ is the minimum number of network configurations required to fulfill all-to-all communication in an $N \times N$ GSEN and $\mathcal{R}_{sc}(N)$ is the minimum number of network configurations required to fulfill all-to-all communication in an $N \times N$ GSEN when the stage control technique is assumed. In Theorem 3.8, we have proven $N \leq \mathcal{R}(N) \leq \mathcal{R}_{sc}(N) = 2^n$. In Theorem 4.8, we have proven $N = \mathcal{R}(N) < \mathcal{R}_{sc}(N) = 2^n$ if $N \equiv 2 \pmod 4$. In Theorem 5.4, we have proven $\mathcal{R}(N) = \mathcal{R}_{sc}(N) = 2^n$ if $k \geq 2, N \equiv 0 \pmod{2^k}, N \not\equiv 0 \pmod{2^{k+1}}$, and $2^{n-1} + 2^{n-k} \leq N \leq 2^n$. In Theorem 5.5, we have proven $\mathcal{R}(20) = 24$.

Before closing this paper, we list $\mathcal{R}(N)$ and $\mathcal{R}_{sc}(N)$ for $N = 4, 6, \ldots, 128$ in Fig. 14. We conjecture that when $N \equiv 4 \pmod 8$, the best way to reduce the number of network configurations used in an all-to-all communication in a GSEN is to use doubly alternating stage control. One can examine $\mathcal{R}(36) \leq 40$ and $\mathcal{R}(44) \leq 48$ by the aid of a computer. We also conjecture that when $N \equiv 8 \pmod{16}$, the best way to reduce the number of network configurations used in an all-to-all communication in a GSEN is to use *quadruply alternating stage control*, meaning that the states of the switches of a stage *alternate between four straight states and four cross states*. The network configuration obtained by quadruply alternating stage control is called a *quadruply alternating configuration* and it can be represented by the number

$$A'' = a''_{n-1}2^{n-1} + a''_{n-2}2^{n-2} + \cdots + a''_0 2^0$$

as follows. Let a''_ℓ denote the states of the switches at stage $n-1-\ell$ such that

- $a''_\ell = 0$ means the states are 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, and so on.
- $a''_\ell = 1$ means the states are 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, and so on.

Obviously, $0 \leq A'' < 2^n$.

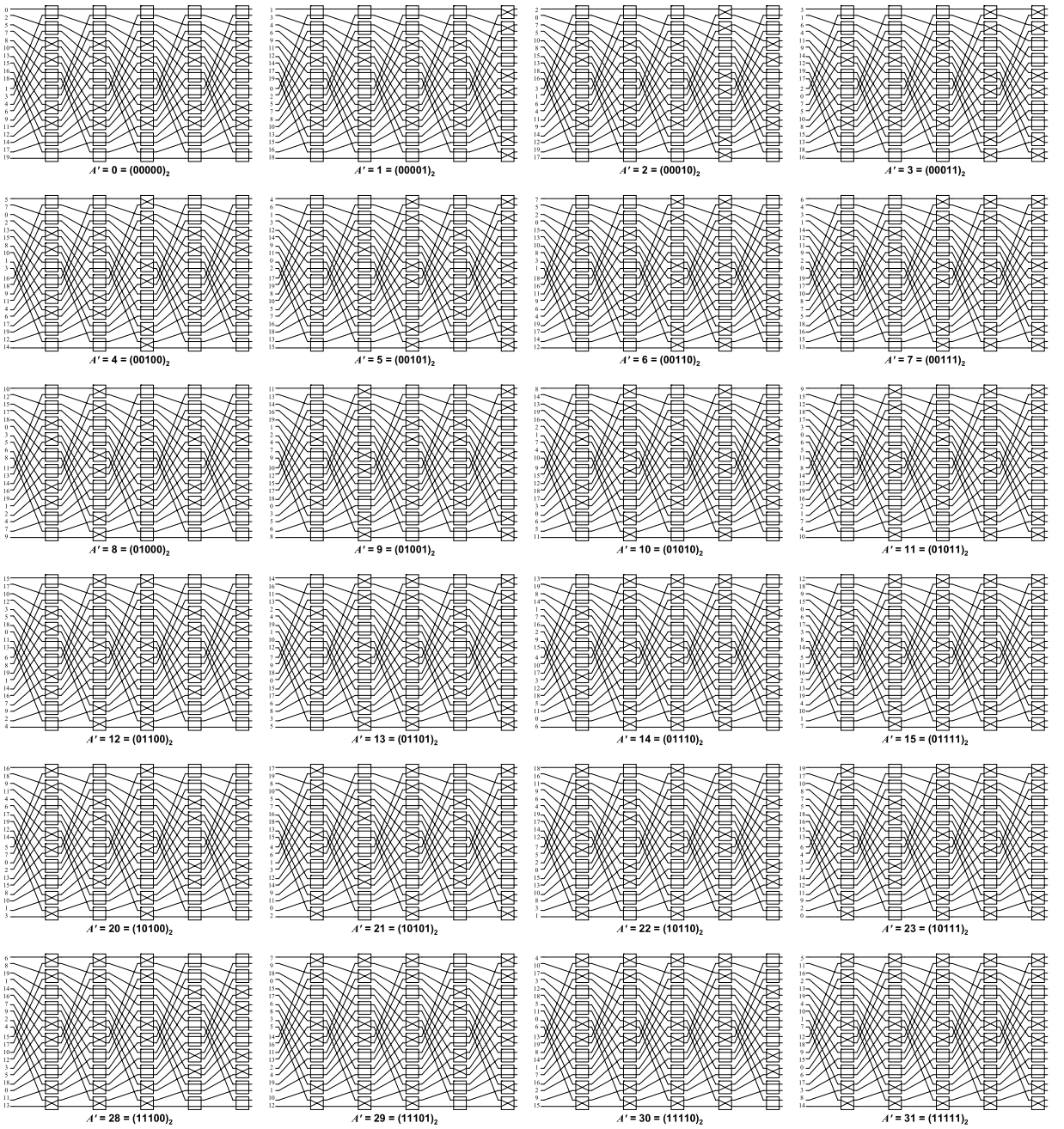

Fig. 13. Fulfill an all-to-all communication in a 20×20 GSEN by using the 24 network configurations in $\mathcal{A}' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31\}$. The destinations of the messages for each network configuration are shown on the left-hand side of the GSEN.

Let $\mathcal{A}'$ denote a set of doubly alternating configurations and let $\mathcal{A}''$ denote a set of quadruply alternating configurations. The following results are obtained by the aid of a computer.

- $\mathcal{R}(36) \leq 40$; by using $\mathcal{A}' = \{0 \sim 3,\ 8 \sim 19,\ 24 \sim 35,\ 40 \sim 43,\ 48 \sim 51, 56 \sim 59\}$.
- $\mathcal{R}(44) \leq 48$; by using $\mathcal{A}' = \{0 \sim 3,\ 8 \sim 19,\ 24 \sim 35,\ 40 \sim 51,\ 56 \sim 63\}$.
- $\mathcal{R}(68) \leq 72$; by using $\mathcal{A}' = \{0 \sim 11,\ 16 \sim 43,\ 48 \sim 63,\ 68 \sim 71,\ 80 \sim 83,\ 100 \sim 104,\ 112 \sim 115\}$.
- $\mathcal{R}(72) \leq 96$; by using $\mathcal{A}'' = \{0 \sim 63,\ 72 \sim 79,\ 88 \sim 95,\ 104 \sim 111,\ 120 \sim 127\}$.
- $\mathcal{R}(76) \leq 88$; by using $\mathcal{A}' = \{0 \sim 7,\ 12 \sim 39,\ 44 \sim 67,\ 80 \sim 91,\ 96 \sim 99,\ 112 \sim 123\}$.
- $\mathcal{R}(84) \leq 96$; by using $\mathcal{A}' = \{0 \sim 11,\ 16 \sim 43,\ 48 \sim 63,\ 68 \sim 71,\ 80 \sim 95,\ 100 \sim 103,\ 112 \sim 127\}$.
- $\mathcal{R}(92) \leq 112$; by using $\mathcal{A}' = \{0 \sim 7,\ 12 \sim 39,\ 44 \sim 71,\ 76 \sim 103,\ 108 \sim 127\}$.

N	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44
$R(N)$	4	6	8	10	16	14	16	18	24	22	32	26	32	30	32	34	≤ 40	38	64	42	≤ 48
$Rsc(N)$	4	8	8	16	16	16	16	32	32	32	32	32	32	32	32	64	64	64	64	64	64

N	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80	82	84	86
$R(N)$	46	64	50	64	54	64	58	64	62	64	66	≤ 72	70	≤ 96	74	≤ 88	78	128	82	≤ 96	86
$Rsc(N)$	64	64	64	64	64	64	64	64	64	64	128	128	128	128	128	128	128	128	128	128	128

N	88	90	92	94	96	98	100	102	104	106	108	110	112	114	116	118	120	122	124	126	128
$R(N)$	128	90	≤ 112	94	128	98	128	102	128	106	128	110	128	114	128	118	128	122	128	126	128
$Rsc(N)$	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128

Fig. 14. Known results of $\mathcal{R}(N)$ and $\mathcal{R}_{sc}(N)$ for $N = 4, 6, \ldots, 128$.

Although we know that $\mathcal{R}(36) \leq 40$, we are unable to prove that $\mathcal{R}(36) \geq 40$. Several open problems can be found in Fig. 14. In particular, we conjecture $\mathcal{R}(36) = 40$, $\mathcal{R}(44) = 48$. Determining $\mathcal{R}(N)$ for all N such that $N \equiv 0 \pmod 4$ is still an open problem.

References

[1] Z. Chen, Z. Liu, Z. Qiu, Bidirectional shuffle-exchange network and tag-based routing algorithm, IEEE Commun. Lett. 7 (3) (2003) 121–123.
[2] C. Chen, J.K. Lou, An efficient tag-based routing algorithm for the backward network of a bidirectional general shuffle-exchange network, IEEE Commun. Lett. 10 (4) (2006) 296–298.
[3] M. Gerla, E. Leonardi, F. Neri, P. Palnati, Routing in the bidirectional shufflenet, IEEE/ACM Trans. Netw. 9 (1) (2001) 91–103.
[4] C.P. Kuruskal, A unified theory of interconnection network structure, Theoret. Comput. Sci. 48 (1986) 75–94.
[5] J.K. Lan, W.Y. Chou, C. Chen, Efficient routing algorithms for the bidirectional general shuffle-exchange network, Discrete Math. Algorithms Appl. 1 (2) (2009) 267–281.
[6] D.H. Lawrie, Access and alignment of data in an array processor, IEEE Trans. Comput. C- 24 (12) (1975) 1145–1155.
[7] S.C. Liew, On the stability if shuffle-exchange and bidirectional shuffle-exchange deflection networkA, IEEE/ACM Trans. Netw. 5 (1) (1997) 87–94.
[8] V.W. Liu, C. Chen, R.B. Chen, Optimal all-to-all personalized exchange in d-nary banyan multistage interconnection networks, J. Comb. Optim. 14 (2007) 131–142.
[9] A. Massini, All-to-all personalized communication on multistage interconnection networks, Discrete Appl. Math. 128 (2) (2003) 435–446.
[10] K. Padmanabhan, Design and analysis of even-sized binary shuffle-exchange networks for multiprocessors, IEEE Trans. Parallel Distrib. Syst. 2 (4) (1991) 385–397.
[11] C. Qiao, L. Zhou, Scheduling switch disjoint connections in stage-controlled photonic banyans, IEEE Trans. Commun. 47 (1) (1999) 139–148.
[12] R. Ramaswami, Multi-wavelength lightwave networks for computer communication, IEEE Commun. Mag. 31 (2) (1993) 78–88.
[13] Y. Yang, J. Wang, Optimal all-to-all personalized exchange in self-routable multistage networks, IEEE Trans. Parallel Distrib. Syst. 11 (3) (2000) 261–274.
[14] Y. Yang, J. Wang, Optimal all-to-all personalized exchange in a class of optical multistage networks, IEEE Trans. Parallel Distrib. Syst. 12 (9) (2001) 567–582.