An Efficient Tag-Based Routing Algorithm for the Backward Network of a Bidirectional General Shuffle-Exchange Network

in Honor of Frank K. Hwang's 65th Birthday

Chiuyuan Chen and Jing-Kai Lou

***Abstract*— This letter considers the problem of designing efficient routing algorithms for the backward network of a bidirectional general shuffle-exchange network (BNBGSEN for short); switch elements in the network are of size $k \times k$. It has been shown in [1] that the algorithm in [5] can be used to obtain (as many as k) backward control tags for a source j to get to a destination i in a BNBGSEN. In this letter, we show that a BNBGSEN has a wonderful property: for each destination i, there are two backward control tags associated with it such that every source j can get to i by using one of the two tags. We use this property to derive an efficient tag-based routing algorithm.**

***Index Terms*— Multistage interconnection network, Omega network, shuffle-exchange, tag-based routing algorithm.**

I. Introduction

THE purpose of this letter is to derive an efficient tag-based routing algorithm for the backward network of a bidirectional general shuffle-exchange network. Throughout this letter, N' denotes the number of inputs and outputs of the network and all the switch elements are of size $k \times k$.

Shuffle-exchange networks have been proposed as a popular architecture for interconnection networks; see [2], [3], [4], [5], [6]. The *general shuffle-exchange operation* on N' terminals $(k \mid N')$ is the permutation π defined by

$$\pi(i) = (ki + \left\lfloor \frac{ki}{N'} \right\rfloor) \bmod N', \quad 0 \le i \le N' - 1.$$

A *shuffle-exchange network* is a network with $N' = k^d$ inputs and outputs and each stage consists of the general shuffle on N' terminals followed by N'/k switch elements.

In a multistage interconnection network, a path from a source to a destination can be described by a sequence of labels that label the successive links on this path. Such a sequence is called a *control tag* [5] or *tag* [1]. The control tag may be used as a header for routing a message: each successive switch element uses the first element of the sequence to route the message, and then discards it.

When N' is a power of k and when the number of stages is exactly $\log_k N'$, the shuffle-exchange network is the Omega network (see [4]) and its control tags depend only on the destination. In [5], Padmanbhan proposed the *general shuffle-exchange network* (GSEN), which allows $N' \neq k^d$ and contains exactly $\lceil \log_k N' \rceil$ stages. Padmanbhan proposed an elegant tag-based routing algorithm for a GSEN and showed that the control tags depend on both the source and the destination when $N' \neq k^d$.

In [1], Chen, Liu and Qiu enhanced the GSEN with bidirectional links. Their reason for the enhancement is that although unidirectional links are widely used, bidirectional links also have many applications as suggested in [2]. A bidirectional GSEN can be divided into two dependent networks: the *forward network* and the *backward network*. The forward (backward) network is from the left-hand (right-hand) side of the network to the right-hand (left-hand) side of the network; thus a request in it is sent from left (right) to right (left). The control tags used in the forward (backward) network are called the *forward* (*backward*) *control tags*.

Since a forward network is a GSEN, Padmanbhan's tag-based routing algorithm can be used in it. As for the backward network, Chen et al. [1] implemented a tag-based routing algorithm (call it CLQ-algorithm for convenience) by using the forward tag inversely. More precisely, CLQ-algorithm first runs Padmanbhan's tag-based routing algorithm to derive the forward control tag; then, it runs another procedure to convert the forward control tag into the backward control tag. If the number of stages is $n + 1$, then CLQ-algorithm takes $O(n)$ time to derive the tag for a source j to get to a destination i and takes $O(N'^2 n)$ time to construct the routing table (a table that contains the backward control tags for routing the $N' \times N'$ pairs of nodes in the backward network).

In this letter, we show that the backward network has a wonderful property: for each destination i, there are two backward control tags associated with it such that every source j can get to i by using one of the two tags. We show that the two tags can be derived in $O(n)$ time. Therefore, it is possible to derive in $O(n)$ time not only a tag for a j to get to i but also the tags for every j to get to i. So, constructing the routing table can be done in $O(N'n)$ time. We summarize below.

time required to	CLQ-algo.	our algo.
find a tag for a j to get to i	$O(n)$	$O(n)$
find the tags for every j to get to i	$O(N'n)$	$O(n)$
construct the routing table	$O(N'^2 n)$	$O(N'n)$

Manuscript received November 3, 2005. The associate editor coordinating the review of this letter and approving it for publication was Aleksander Kolarov. This research was partially supported by the National Science Council of the Republic of China under the grant NSC94-2115-M-009-006.

The authors are with the Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: cychen@mail.nctu.edu.tw).

Digital Object Identifier 10.1109/LCOMM.2006.04007.

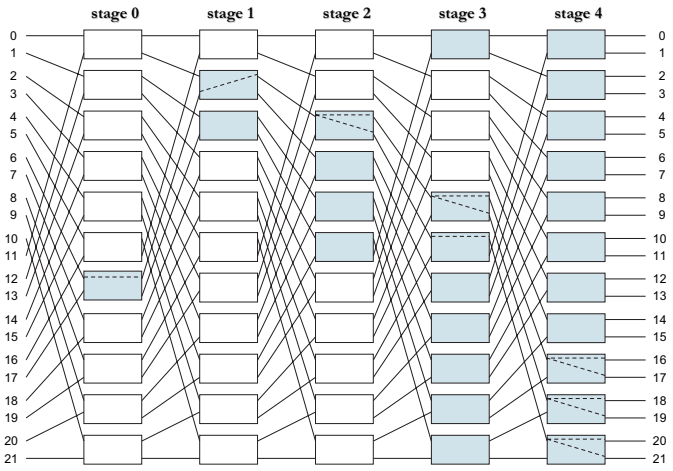

Fig. 1. GSEN(2,11,5). The switch elements that can get to $i = 6$ are highlighted; if $j \geq 16$, then j can get to $i = 6$ using the tag 0 0 0 1 0.

II. Bidirectional GSEN and Conventions

The following definition was given in [1]. A *bidirectional general shuffle-exchange network* GSEN$(k, r, n+1)$ is a GSEN with bidirectional links. The switch elements are aligned in $n+1$ stages, labelled $0, 1, \cdots, n$. Each stage consists of r switch elements, labelled $0, 1, \cdots, r-1$. And each switch element is a $k \times k$ bidirectional crossbar. For example, the network in Figure 1 is GSEN(2,11,5).

Note that in GSEN$(k, r, n+1)$, there are a total of $N' = k \times r$ ports on each side of a stage, labelled $0, 1, \cdots, N'-1$. The parameters k, r and n satisfy the following equation:

$$\lceil \log_k (k \cdot r) \rceil = \lceil \log_k N' \rceil = n + 1.$$

Throughout this letter, in a stage, the switch element labelled 0 is considered to be the successive switch element of the switch element labelled $r-1$. Also, terminal i (j) is assumed on the left-hand (right-hand) side of the network.

III. Our Algorithm

In this section, we will propose an algorithm to compute the two backward control tags $s_n\ s_{n-1}\ \cdots\ s_0$, $s'_n\ s'_{n-1}\ \cdots\ s'_0$ and a critical value $v(i)$ associated with a given i. We will prove that if $j < v(i)$, then j can get to i by using the tag $s_n\ s_{n-1}\ \cdots\ s_0$ and if $j \geq v(i)$, then j can get to i by using the tag $s'_n\ s'_{n-1}\ \cdots\ s'_0$.

The following observations are crucial to our algorithm: At stage 0, only one switch element can get to i. At stage 1, exactly k switch elements can get to i and these switch elements are *consecutive*. At stage 2, exactly k^2 switch elements can get to i and these switch elements are *consecutive*. In general, at stage ℓ, $0 \leq \ell \leq n-1$, exactly k^ℓ switch elements can get to i and these switch elements are *consecutive*. Clearly, at stage n (the last stage), all the r switch elements can get to i. Since at stage ℓ the switch elements that can get to i are consecutive, we only need to store the label of the first one. Let C_ℓ denote this label. Clearly, we have $C_\ell = i \times k^\ell \pmod r$. A critical value $v(i)$ associated with i is defined to be $v(i) = C_n \times k$. The following is our algorithm.

BACKWARD-CONTROL-TAGS.

Input: i on the left-hand side of a bidirectional GSEN$(k, r, n+1)$.

Output: The critical value $v(i)$ and the two backward control tags $s_n\ s_{n-1}\ \cdots\ s_0$ and $s'_n\ s'_{n-1}\ \cdots\ s'_0$ associated with i.

1. /* Compute $C_0, C_1, \cdots, C_n$. */
   for $\ell = 0$ to n do
   $C_\ell \leftarrow i \times k^\ell \pmod r$;
2. /* Compute the critical value $v(i)$. */
   $v(i) \leftarrow C_n \times k$;
3. /* Compute $F_0, F_1, \cdots, F_n$. */
   if $(r - C_{n-1}) \times k \geq r$
   then
   begin
   for $\ell = 0$ to $n-1$ do $F_\ell \leftarrow 0$;
   $F_n \leftarrow 1$;
   end
   else
   for $\ell = 0$ to n do
   if $C_\ell + k^\ell > r$ then $F_\ell \leftarrow 1$ else $F_\ell \leftarrow 0$;
4. /* Compute the tag $s'_n\ s'_{n-1}\ \cdots\ s'_0$. */
   $s'_0 \leftarrow \lfloor \frac{i}{r} \rfloor$;
   for $\ell = 1$ to n do $s'_\ell \leftarrow \left\lfloor \frac{k \times C_{\ell-1}}{r} \right\rfloor$;
5. /* Compute the tag $s_n\ s_{n-1}\ \cdots\ s_0$. */
   for $\ell = 0$ to n do $s_\ell \leftarrow s'_\ell + F_\ell \pmod k$;

END-of-BACKWARD-CONTROL-TAGS.

Again, take Figure 1 as an example. Suppose $i = 6$. Then $C_0 = 6$, $C_1 = 1$, $C_2 = 2$, $C_3 = 4$, $C_4 = 8$, $v(i) = 16$, $F_0 = 0$, $F_1 = 0$, $F_2 = 0$, $F_3 = 0$, $F_4 = 1$. Thus

$$s_4\ s_3\ s_2\ s_1\ s_0 = 1\ 0\ 0\ 1\ 0, \quad s'_4\ s'_3\ s'_2\ s'_1\ s'_0 = 0\ 0\ 0\ 1\ 0.$$

It is not difficult to verify that: if $j < 16$, then j can get to 6 by using the tag **1 0 0 1 0**; if $j \geq 16$, then j can get to 6 by using the tag **0 0 0 1 0**.

Recall that there are a total of N' ports on each side of a stage, labelled $0, 1, \cdots, N'-1$. A port R consists of two parts: the number y of the switch element where R is located, and the sub port number z in the switch element where R is located; see [1]. R and y and z satisfy $R = k \cdot y + z$. The following result was proved in [1].

Lemma 1: [1] Suppose port u of stage $\ell - 1$ and port v of stage ℓ are connected by a link, where $u = k \cdot y_1 + z_1$ and $v = k \cdot y_2 + z_2$. Then $z_2 = \lfloor \frac{k \cdot u}{N'} \rfloor$.

Thus we have

Lemma 2: Let u, v, y_1, z_1, y_2, z_2 be defined as in Lemma 1 and consider the switch elements labelled y_1 and y_2. Then the backward control tag for y_2 to get to y_1 (or for y_2 to get to u) is z_2; moreover, $z_2 = \lfloor \frac{u}{r} \rfloor$.

Proof: Clearly, the tag is z_2. Since $N' = k \times r$, by Lemma 1, $z_2 = \lfloor \frac{u}{r} \rfloor$. ∎

We now prove that

Lemma 3: If $j = v(i)$, then j can get to i by using the tag $s'_n\ s'_{n-1}\ \cdots\ s'_0$.

Proof: If $j = v(i)$, then j can get to i via switch elements labelled C_n, C_{n-1}, $\cdots$, C_0. For each ℓ, $1 \leq \ell \leq n$, C_ℓ is linked to $C_{\ell-1}$ via sub port 0 of $C_{\ell-1}$. Sub port 0 of $C_{\ell-1}$ is

port u of $C_{\ell-1}$, where $u = k \times C_{\ell-1}$. Thus by Lemma 2, the tag for C_ℓ to get to $C_{\ell-1}$ is $\left\lfloor \frac{k \times C_{\ell-1}}{r} \right\rfloor$. Also by Lemma 2, the tag for C_0 to get to i is $\lfloor \frac{i}{r} \rfloor$. In Step 3, we set $s'_0 = \lfloor \frac{i}{r} \rfloor$ and $s'_\ell = \left\lfloor \frac{k \times C_{\ell-1}}{r} \right\rfloor$, for $\ell = 1, 2, \cdots, n$. Hence the lemma. ∎

Lemma 4: If $j > v(i)$, then j can get to i by using the tag $s'_n \ s'_{n-1} \ \cdots \ s'_0$.

Proof: Note that $k^n < N' \leq k^{n+1}$. Set $d = j - v(i)$ for easy writing. Then $0 < d \leq N' - 1$. Thus $0 < \frac{d}{k^{n-\ell+1}} \leq \frac{N'-1}{k^{n+1}} \leq \frac{N'-1}{N'} k^\ell < k^\ell$ and therefore $0 \leq \lfloor \frac{d}{k^{n-\ell+1}} \rfloor < k^\ell$. Note that j can get to i via switch elements labelled $C_n + \lfloor \frac{d}{k} \rfloor$, $C_{n-1} + \lfloor \frac{d}{k^2} \rfloor$, $C_{n-2} + \lfloor \frac{d}{k^3} \rfloor$, $\cdots$, $C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$, $\cdots$, $C_1 + \lfloor \frac{d}{k^n} \rfloor$, $C_0 + \lfloor \frac{d}{k^{n+1}} \rfloor$. The connection of a GSEN ensures that if C_ℓ, $1 \leq \ell \leq n$, is connected to $C_{\ell-1}$ via sub port z_2, then $C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ is connected to $C_{\ell-1} + \lfloor \frac{d}{k^{n-\ell+2}} \rfloor$ via sub port z_2. By Lemma 2, the tag for $C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ to get to $C_{\ell-1} + \lfloor \frac{d}{k^{n-\ell+2}} \rfloor$ is z_2; by Lemma 3, $z_2 = s'_\ell$. Note that $0 < \frac{d}{k^{n+1}} \leq \frac{N'-1}{N'} < 1$. Thus $C_0 + \lfloor \frac{d}{k^{n+1}} \rfloor = C_0$. By Lemma 3, the tag for C_0 to get to i is s'_0. Hence the lemma. ∎

Lemma 5: If $j < v(i)$ and $(r - C_{n-1}) \times k \geq r$, then j can get to i by using the tag $s_n \ s_{n-1} \ \cdots \ s_0$.

Proof: Set $d = j - v(i) + N'$ for easy writing. Then j can get to i via switch elements labelled $C_n + \lfloor \frac{d}{k} \rfloor - r$, $C_{n-1} + \lfloor \frac{d}{k^2} \rfloor$, $C_{n-2} + \lfloor \frac{d}{k^3} \rfloor$, $\cdots$, $C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$, $\cdots$, $C_1 + \lfloor \frac{d}{k^n} \rfloor$, $C_0 + \lfloor \frac{d}{k^{n+1}} \rfloor$. The connection of a GSEN ensures that if C_n is connected to C_{n-1} via sub port z_2, then $C_n + \lfloor \frac{d}{k} \rfloor - r$ is connected to $C_{n-1} + \lfloor \frac{d}{k^2} \rfloor$ via sub port $z_2 + 1 \pmod k$. By Lemma 2, the tag for $C_n + \lfloor \frac{d}{k} \rfloor - r$ to get to $C_{n-1} + \lfloor \frac{d}{k^2} \rfloor$ is $z_2 + 1 \pmod k$. By Lemma 3, $z_2 = s'_n$. In our algorithm, we set $F_n = 1$ and set $s_n = s'_n + F_n \pmod k$. Thus $s_n = z_2 + 1 \pmod k$. Again, the connection of a GSEN ensures that if C_ℓ, $1 \leq \ell \leq n-1$, is connected to $C_{\ell-1}$ via sub port z_2, then $C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ is connected to $C_{\ell-1} + \lfloor \frac{d}{k^{n-\ell+2}} \rfloor$ via sub port z_2. By Lemma 2, the tag for $C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ to get to $C_{\ell-1} + \lfloor \frac{d}{k^{n-\ell+2}} \rfloor$ is z_2. By Lemma 3, $z_2 = s'_\ell$. In our algorithm, we set $F_\ell = 0$ and set $s_\ell = s'_\ell + F_\ell \pmod k$. Thus $s_\ell = z_2$. Note that $0 < \frac{d}{k^{n+1}} \leq \frac{N'-1}{N'} < 1$. Thus $C_0 + \lfloor \frac{d}{k^{n+1}} \rfloor = C_0$. By Lemma 3, the tag for C_0 to get to i is s'_0. In our algorithm, we set $F_\ell = 0$ and set $s_0 = s'_0 + F_0 \pmod k$. Thus $s_0 = s'_0$. We now have this lemma. ∎

Lemma 6: If $j < v(i)$ and $(r - C_{n-1}) \times k < r$, then j can get to i by using the tag $s_n \ s_{n-1} \ \cdots \ s_0$.

Proof: Set $d = j - v(i) + N'$ for easy writing. Then j can get to i via switch elements labelled L_n, L_{n-1}, $\cdots$, L_ℓ, $\cdots$, L_1, L_0, where

$$L_n = C_n + \left\lfloor \frac{d}{k} \right\rfloor - r$$

and for $\ell = n-1, n-2, \cdots, 0$,

$$L_\ell = \begin{cases} C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor & \text{if } C_\ell + k^\ell \leq r, \\ C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor - r & \text{if } C_\ell + k^\ell > r. \end{cases}$$

The connection of a GSEN ensures that if C_n is connected to C_{n-1} via sub port z_2, then L_n is connected to L_{n-1} via sub port $z_2 + 1 \pmod k$. By Lemma 2, the tag for L_n to get to L_{n-1} is $z_2 + 1 \pmod k$. By Lemma 3, $z_2 = s'_n$. Note that $C_n + k^n > r$. Thus our algorithm sets $F_n = 1$. Since our algorithm sets $s_n = s'_n + F_n \pmod k$, clearly $s_n = z_2 + 1 \pmod k$. Again, the connection of a GSEN ensures that if C_ℓ, $1 \leq \ell \leq n-1$, is connected to $C_{\ell-1}$ via sub port z_2, then L_ℓ is connected to $L_{\ell-1}$ via sub port z_2 if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ and via sub port $z_2 + 1 \pmod k$ if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor - r$. Thus by Lemma 2, the tag for L_ℓ to get to $L_{\ell-1}$ is z_2 if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ and is $z_2 + 1 \pmod k$ if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor - r$. By Lemma 3, $z_2 = s'_n$. Our algorithm sets $F_\ell = 0$ if $C_\ell + k^\ell \leq r$ (i.e., if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$), sets $F_\ell = 1$ if $C_\ell + k^\ell > r$ (i.e., if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor - r$) and sets $s_\ell = s'_\ell + F_\ell \pmod k$. Thus $s_\ell = z_2$ if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor$ and $s_\ell = z_2 + 1 \pmod k$ if $L_\ell = C_\ell + \lfloor \frac{d}{k^{n-\ell+1}} \rfloor - r$. Note that $0 < \frac{d}{k^{n+1}} \leq \frac{N'-1}{N'} < 1$. Thus $L_0 = C_0$. By Lemma 3, the tag for L_0 to get to i is s'_0. Note that $C_0 + k^0 \leq r$. Thus our algorithm sets $F_0 = 0$ and set $s_0 = s'_0 + F_0 \pmod k$. Thus $s_0 = s'_0$. We now have this lemma. ∎

The following is our main result:

Theorem 7: If $j < v(i)$, then j can get to i by using the tag $s_n \ s_{n-1} \ \cdots \ s_0$; if $j \geq v(i)$, then j can get to i by using the tag $s'_n \ s'_{n-1} \ \cdots \ s'_0$. Moreover, it takes $O(n)$ time to compute $v(i)$ and the two tags.

Proof: Since k^ℓ can be computed from $k^{\ell-1}$ by $k^\ell = k \cdot k^{\ell-1}$, it takes $O(1)$ time to compute k^ℓ. So, it takes $O(n)$ time to compute $v(i)$ and the two tags. This theorem now follows from Lemma 3, Lemma 4, Lemma 5 and Lemma 6. ∎

The following is the routing table for GSEN(2, 11, 5), which is generated by a computer program implemented from our algorithm.

i	$v(i)$	$s_4 \ldots s_0$	$s'_4 \ldots s'_0$	i	$v(i)$	$s_4 \ldots s_0$	$s'_4 \ldots s'_0$
0	0	10000	00000	11	0	10001	00001
1	10	01000	10000	12	10	01001	10001
2	20	11000	01000	13	20	11001	01001
3	8	10100	00100	14	8	10101	00101
4	18	01100	10100	15	18	01101	10101
5	6	00010	11100	16	6	00011	11101
6	16	10010	00010	17	16	10011	00011
7	4	11010	01010	18	4	11011	01011
8	14	00110	11010	19	14	00111	11011
9	2	01110	10110	20	2	01111	10111
10	12	11110	01110	21	12	11111	01111

REFERENCES

[1] Z. Chen, Z. Liu and Z. Qiu, "Bidirectional shuffle-exchange network and tag-based routing algorithm," *IEEE Commun. Lett.*, vol. 7, no. 3, pp. 121-123, Mar. 2003.

[2] M. Gerla, E. Leonardi, F. Neri, and P. Palnati, "Routing in the bidirectional shufflenet," *IEEE/ACM Trans. Networking*, vol. 9, no. 1, pp. 91-103, Feb. 2001.

[3] F. K. Hwang, "The mathmatical theory of nonblocking switching networks," *Series on Applied Mathematics*, vol. 15, ch. 1, pp. 12-22, 2004.

[4] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. 24, no. 12, pp. 1145-1155, Dec. 1975.

[5] K. Padmanabham, "Design and analysis of even-sized binary shuffle-exchange networks for multiprocessors," *IEEE Trans. Parallel and Distrib. Syst.*, vol. 2, no. 4, pp. 385-397, Oct. 1991.

[6] R. Ramaswami, "Multi-wavelength lightwave networks for computer communication," *IEEE Commun. Mag.*, vol. 31, no. 2, pp. 78-88, Feb. 1993.