論文 89-26-6-3

On a Distributed Routing Control Scheme in Multistage Baseline Networks

(다단계 베이스라인 네트워크에서의 분산 라우팅 제어 방법에 관한 연구)

孫 有 翼*, 安 光 善**

(Yoo Ek Son and Gwang Seon Ahn)

要 約

본 논문은 다단계 베이스라인 네트워크에서 네트워크의 제어를 스위칭 소자들에 분산시키는 라우팅 방법에 대해서 다룬다. 여기서 사용된 라우팅 방법은 목적지 주소의 이진표현을 라우팅 택으로 사용하는 목적지 택 라우팅 방법을 채택하였다. 주어진 네트워크의 구조적 특성을 이용하여, 하나의 시작지 주소로 부터 임의의 목적지 주소로 전송이 가능한 분산 알고리듬을 제안하였다. 제안된 알고리듬의 성능분석은 컴퓨터 시뮬레이션을 통하여 수행되었으며 그 결과를 시간 복잡도 면에서 다른 방법과 비교하였다.

Abstract

This paper presents a distributed routing scheme for allowing network control to be distributed through the switching elements of a multistage baseline network. The routing technique we use here is the destination tag scheme which uses the binary representation of a destination address as the routing tag. With the topological properties of the network, we introduce a distributed algorithm which allows any connection from a source to arbitrary number of destinations. Also, the performance of the proposed method is evaluated by computer simulation, and the results of the method is compared to other schemes in terms of time complexity.

*正會員, 啓明大學校 電子計算學科
(Dept. of Computer Science, Keimyung Univ.)
**正會員, 慶北大學校 電子計算機工學科
(Dept. of Computer Eng., Kyungpook Nat'l Univ.)
接受日字 : 1988年 9月 21日

I. Introduction

With the advent of VLSI development technologies, recently, the interconnection networks are the increasingly important area of research as a major problem in designing large-scale parallel/distributed systems which can satisfy the needs

for powerful computing functions with increased performance and improved reliability. Therefore, to design a proper interconnection network has long been studied as one of the major issues in developing a multiprocessor system because overall system performance relies highly upon the network[1]. It has been known that multistage interconnection networks, MINs, are cost-effective in providing high-bandwidth communication in multiprocessors[4,10]. A multistage network consists of more than one stage of switching elements and is usually capable of connecting an arbitrary source to an arbitrary destination. Several networks have been proposed[3,5,6,7,8,12]. Interconnection network is a set of interconnection functions which are realized by properly setting control of the switching elements.

The routing control structure of a network determines how the states of the switching elements will be set and depends on the network topology and the control strategy. There exist two types of control strategies in multistage networks, individual stage control and individual switching element control[1,4,9]. Distributed routing control is the settings of the individual switching elements determined by routing information contained within the routing tag. For baseline network, two types of routing techniques are available: recursive routing and destination tag routing[1,5,7].

In this paper we intended to study a distributed routing control scheme based on the individual switching element control scheme, so that network control can be distributed through the switching elements, and hence, any source can broadcast to any set of destinations by appropriately setting the switching elements.

II. Network Configuration

The performance of an interconnection network is typically determined by its configuration, which consists of three parameters such as the number of communication paths of each switching element, the number of stages, and the interconnection links between stages.

Here, we use a baseline network[7] as a reference model for representing other existing multistage interconnection networks. In the network, there are $\log_2 N$ stages of the switching elements. The switching elements are arranged into $\log_2 N$ stages of N/2 switching elements and the stages are labeled in a sequence from $\log_2 N$ -1 to 0, with 0 for the rightmost stage. Each switching element is basically a crossbar switch which has two input lines and two output lines. As valid states of a switching element, we denote four states; direct, exchange, lower broadcast and upper broadcast connection as illustrated in Fig.2.

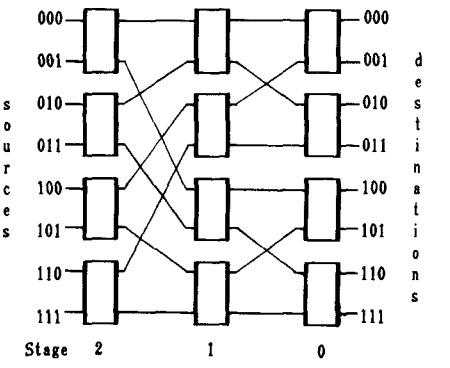


Fig.1. A baseline network, with N=8.

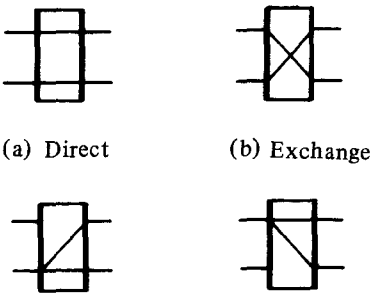


Fig.2. Four valid states of a switching element.

Assume that the inputs connected to the switching elements in the leftmost stage, stage m-1, represent the sources, and the outputs from the switching elements in the rightmost stage, stage 0, represent the destinations. Each switching element is named by a codeword of $p_{m-1} \ldots p_2\ p_1$ which is the binary representation of its location

in the stage. And each link is named by a codeword of $p_{m-1} \cdots p_1 p_0$, which is coded by the following manner; the last bit, p_0, is equal to 0 if the link is connected to an upper link of the switching element, and p_0 is equal to 1 if the link is connected to a lower link. The interconnection patterns between stages of a baseline network can be defined by applying the interconnection function

$$F(i)\ [p_{m-1}\ p_{m-2} \cdots p_1\ p_0] = p_{m-1} \cdots p_{i+1}\ p_0\ p_i\ p_{i-1} \cdots p_1$$

A switching element in stage i is mapped to two switching elements in stage i-1 by the following mapping rules:

$$M(0,i)\ [p_{m-1}\ p_{m-2} \cdots p_1)_i] = (p_{m-1} \cdots p_{m-1}\ p_{m-i-1} \cdots p_2)_{i-1},$$
for upper link $(p_{m-1}\ p_{m-2} \cdots p_1\ 0)_{i-1}$, $0 \leqslant i \leqslant m-1$

$$M(1,i)\ [p_{m-1}\ p_{m-2} \cdots p_1)i] = (p_{m-1} \cdots p_{m-i}\ 1\ p_{m-i-1} \cdots p_2)_{i-1},$$
for lower link $(p_{m-1}\ p_{m-2} \cdots p_1\ 1)_{i-1}$, $0 \leqslant i \leqslant m-1$

where $M(0,i)$ and $M(1,i)$ by upper and lower links, respectively, represent the interconnections by mapping a switching element in stage i to two switching elements in stage $i+1$, one element per link out of the switching element in stage i.

Ⅲ. Distributed Routing Schemes

Each switching element is individually controlled through the use of routing tags when routing data from a source to a set of destinations. In one-to-one connections, the network involve just the direct and exchange switch settings and use the m-bit destination tag for data routing[9] In one-to-many connections, involving two more states; upper broadcast and lower broadcast, in addition to the direct and exchange. In this case, 2m-bits are used for routing tags: the routing and broadcast informations which are specified by {R,B}, where $R=r_{m-1} \cdots r_1\ r_0$ and $B=b_{m-1} \cdots b_1\ b_0$[8].

The destination tag routing scheme used in this paper is the method that the binary representation of the destination address is used as a routing tag. And it is very suitable for a distributed control because the destination tag routing scheme will connect the only path available between a source and destinations. Assume the source link and the destination link are $S = s_{m-1} \cdots s_1\ s_0$ and $D= d_{m-1} \cdots d_1\ d_0$, respectively. Starting from the source S, the switching element in the first stage (stage 2) to which S is connected is set to switch S to the upper link if $d_{m-1} = 0$ and the lower link if $d_{m-1} = 1$. The second switching element in stage $m-2$ (stage 1) is again set to switch S according to the states of d_{m-2}

At first, consider the stage control method[8] The routing tage consists of two informations: routing information and broadcast information. The routing information can be obtained as any one of the desired destinations which is equal to the destination tag. Broacast information can be obtained by $D_f \oplus D_l$, where D_f and D_l are any two destinations that differ by j bits[5,7,8,9]. To specify a broadcast, set b_i to 1 if the switching element in stage i is to be set to one of the broadcast states, and set b_i to 0 if the switching element is to interpret the R information in {R,B} as a normal routing tag, where b_i is a bit included in broadcast information. Here, when the number of the destinations to be connected is k, where $k= 2^j$, the Hamming distance between the two output addresses (D_f and D_l) must be equal to j. Thus, in stage control scheme, a grouping technique is needed for sending the data sequentially when the desired destinations can not satisfy the condition. These schemes are shown in Table 1. As an example which satisfies the condition of the stage control method, when assuming a set of destination addresses is (010, 011, 110, 111), we can see the condition is satisfied because they differ in at most two bit positions. To compute the (R,B), R=010 can be obtained by D_f, and B=101, by $D_f \oplus D_l$. The routing scheme is illustrated in Fig.3.

Next, consider the distributed control algorithm which is to allow network control to be distributed through the switching elements of a network. The distributed control algorithm proposed in this paper makes a source connect to arbitrary destinations. As a basic principle for the algorithm, we establish the following observations from the topological properties of a baseline network.

Observation 1: The physical names of switching elements and links in the network can be assigned

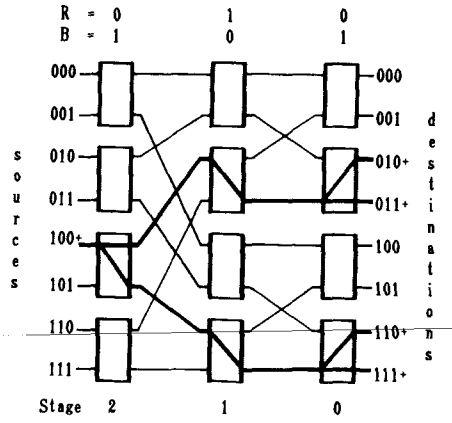


Fig.3. Broadcast path from input address 4 to output addresses 2, 3, 6, and 7.

Table 1. Algorithm_1: stage control.

Step 1:
read the number of stages and source address.
count the number of destinations.

Step 2:
permute and sort the destinations.
make groups by 2^j.

Step 3:
decide the internal conditions of reach group.
make the groups.
check validity of the elements in each groups.
generate the {R,B} information.

Step 4:
transmit each data group to the destinations by the number of groups sequentially.

by $p_{m-1} \cdots p_2\ p_1$ and $p_{m-1} \cdots p_2\ p_1\ p_0$, repectively.

Observation 2: The routing tag in the baseline network will have m bits and is as the following form

$$d_{m-1}\ d_{m-2} \cdots d_1\ d_0$$

It is the destination address. d_i is used to set the switching elements in stage i of the network by selecting the destination.

Observation 3: We can compute the set of addresses between a source and destinations by using the fact that there must exist a routing path from a source to a destination. For example, if a source address is 010, and a destination address is 110 in binary for N=8, the path between the two can be represented as a set of input/output links of each stage according to the following procedures:

1) Obtain the destination address, 110, as the routing tag, because the destination tag routing scheme is used in this network.
2) i_port_2=010, because the input link connected to the stage m-1 (stage 2) is the source address (010). Thus, 0_port_2=011 can be obtained by the tag bit, 1, which causes the link connect to a lower link. ($m=\log_2 N$, network size N=8)
3) And next, i_port_1=101 by applying the o_port_2 address (011) to the interconnection function $F(i)\ [p_{m-1}\ p_{m-2} \cdots p_1\ p_0] = p_{m-1} \cdots p_{i+1}\ p_0\ p_i\ p_{i-1} \cdots p_1$. Also, it is the input link address of the next stage (stage 1). If repeating the procedure 2) and 3) by the number of stages, for a given example, we obtain

i_port_2 = 010
o_port_2 = 011, by the first tag '1'
i_port_1 = 101, by F(2) [011] = 101
o_port_1= 101, by the second tag '1'

i_port_0 = 110, by F(1) [101] = 110
o_port_0 = 110, by the last tag '0'

where i_port_j and o_port_j are the addresses of input and output ports of a switching element in stage j, respectively, and $0 \leqslant j \leqslant m-1$. Therefore, we can obtain a full path, path {2 3 5 4 6 6}, in one-to-one connections. When extending the method to one-to-many connections, the paths for every destination can be represented by a path table as shown in Examples. In Fig.4, the routing procedure for Observation 2 is shown.

Observation 4: Given a link address, we can compute the physical name of the switching element to which the link is connected. For example, an address of arbitrary switching element and the number of the stage where it is involved can be obtained by the path {2 3 5 5 6 6},

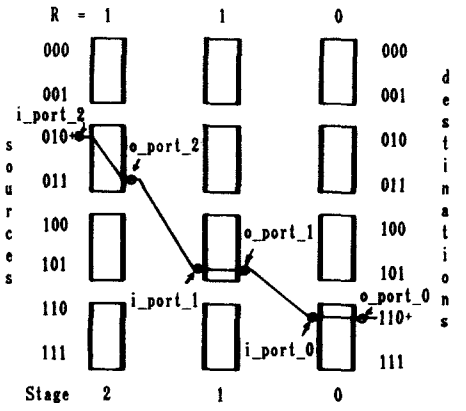


Fig.4. The routing procedure for Observation 3.

Therefore, we can find the following informations.

1) The three switching elements are active from a source to a destination, {2 6} connection path, and the setting states of the three switching elements are 'exchange,' 'direct', and 'direct', respectively.
2) If appyling all the input links 010, 101, and 110 to be connected to the switching elements to baseline naming function, $N(p_{m-1} \ldots p_0)$ $p_{m-1} \ldots p_1$, N(010) =01 1, N(101)=10 2, and N(110)=11 3, respectively.

Therefore, if a switching element can be represented by SE[i,j], where i is the physical name and j is the number of stage, the control signals can be generated as follows:

SE [1,2]=1
SE [2,1]=0
SE [3,0]=0

IV. Simulation and Examples

We have constructed the simulation system to present the distributed routing scheme proposed in this paper. It is composed of six modules including destination tag generation, path table generation, interconnection functions, address computation of the link connected to a switching element, SE physical name computation and control signals generation.

Table 2. Algorithm 2: distribyted control

Inputs:
. number of stage, m
. source address, s
. number of destinations, d_n

Outputs:
. distributed control signals

Step 1:
generate the destination tag.
(d_tag[i,j])

Step 2:
generate path table (path_1).
allocate the next link.
compute the next link address.

Step 3:
rearrange the path table to avoid duplicating links in the same stage.
generate path_2 table.

Step 4:
detect the active switching elements and generate the control signals.
compute the location of the active switching element.
compute stage no. containing the switching element.

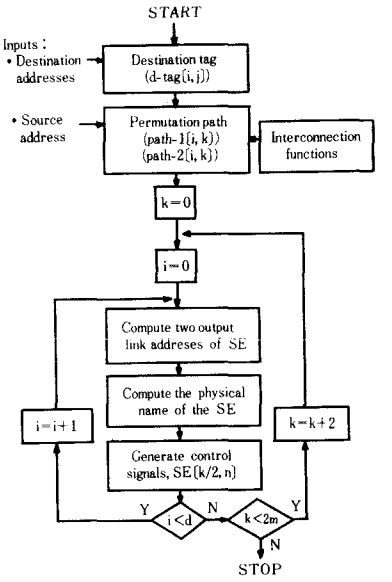


Fig.5. Block diagram of the simulation system.

The simulation system generates routing tags, a path table, an array of switching element control signals, and a set of addresses from a source to destinations using the interconnection functions between stages and the states of each switching element in the network. The block diagram of simulation system is shown in Fig.5. The following Examples 1 and 2 are the cases in one-to-one connection and broadcasting respectively. As a result of the simulation, we show the case that it is not possible to send data at a time with the stage control scheme, but possible with the distributed scheme as illustrated in Fig.6. These properties make routings flexible for connecting to the desired destinations regardless of the restriction[5,7,8]. The four states of a switching element are indicated by

'*' for Inactive
'0' for Direct connection
'1' for Exchange connection
'2' for Lower broadcast connection
'3' for Upper broadcast connection

[Example 1] One-to-one connection:

- Source address: 101
- Destination address: 010

- Path table:

5 3 2 2 2 2

- Control singals:

SE[0,2]=*	SE[0,1]=*	SE[0,0]=1
SE[1,2]=*	SE[1,1]=1	SE[1,0]=*
SE[2,2]=*	SE[2,1]=*	SE[2,0]=*
SE[3,2]=1	SE[3,1]=*	SE[3,0]=*

[Example 2] Broadcasting:

- Source address: 6
- Destination addresses:0, 1, 4 and 7

- Path table:

6	6	3	2	1	0
–	7	7	6	5	1
–	–	–	7	7	4
–	–	–	–	–	7

- Control signals:

SE[0,2]=*	SE[0,1]=*	SE[0,0]=2
SE[1,2]=*	SE[1,1]=1	SE[1,0]=*
SE[2,2]=*	SE[2,1]=*	SE[2,0]=1
SE[3,2]=3	SE[3,1]=2	SE[3,0]=0

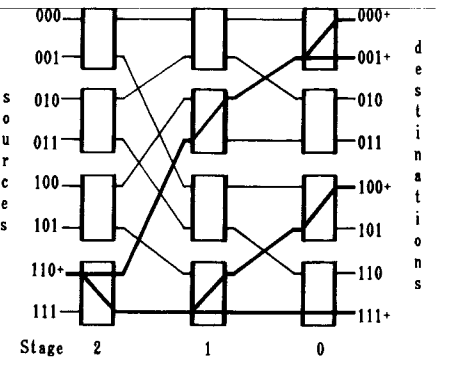


Fig.6. Broadcast path from input address 6 to output addresses 0, 1, 4, and 7.

V. Analysis and Conclusion

To evaluate the time complexity of the distributed routing algorithm proposed in this study, first of all, we describe the notations to use in this analysis as follow:

N : network size
m : the number of stage ($m=\log_2 N$)
s : the address of a source
d_n : the number of destinations
j : Hamming distance between D_f and D_l
k : the number of destinations in a group ($k=2^j$)
g : the number of groups

For stage control scheme as shown in Table 1, in Step 1, it will be $d_n \times \log_2 N$ for obtaining the destnations with binary representation. In Step 2, for permuting them and for sorting the permuted destinations, $2d_n \times \log_2 N$ is needed. In Step 3, $2d_n \times \log_2 N$ for checking the condition of grouping, $d_n \times \log_2 N$ for checking the validity of the internal elements in each group, and $g \times 2\log_2 N$

for generating the routing and broadcast informations of the groups will be obtained. Finally, in Step 4, the process for data transmission of the groups through the network can be composed into two phases: setup time (T_{setup}) and data transfer time (T_{data}). The T_{setup} can be $T_{SE} \times \log_2 N + T_g$, where T_{SE} is the propagation delay time of a switching element and T_g is the time for grant signal. Next, the data transfer time, T_{data}, will be $g \times \log_2 N$ for sending the groups through the network. Thus, the total time will be

$$6d_n \times \log_2 N + 2(\log_2 N)^2 + 3g \times \log_2 N.$$

For distributed control as shown in Table 2, in Step 1, it will be $d_n \times \log_2 N$. In Step 2, the time will be obtained by the number of destinations, d_n, and the number of stages, $\log_2 N$. Thus, it will be $d_n \times \log_2 N$. In Step 3, it is $2d_n \times \log_2 N$, because the path_2 have a form of matrix $[d_n \times 2m]$.

Finally, the time to perform the procedure of the step 4 will be $2d_n \times \log_2 N$. Thus, the total time will be

$$6d_n \times \log_2 N.$$

As a concluding remark, we have presented a distributed routing algorithm based on the individual switching element control method in multistage baseline networks. Through the analysis of time complexity, we show that the proposed method is more effective in comparison with that of stage control scheme. Also, we illustrated the validity of the routing algorithm through the simulation. With the results, the proposed algorithm will be applicable for any connection between a source and arbitrary number of destinations.

References

[1] T.Y. Feng, "A Survey of interconnection networks," *IEEE Computer*, pp. 12-27, Dec. 1981.

[2] H.J. Siegel, "Interconnection networks for SIMD machines," *IEEE Computer* pp. 110-118, June 1979.

[3] M.C. Peace, "The indirect binary n-cube microprocessor array," *IEEE Trans. on Computers*, vol. C-26, May 1977.

[4] J.H. Patel, "Performance of processor-memory interconnection for multiprocessors," *IEEE Trans. on Computers*, vol. C-30, Oct. 1981.

[5] D.H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. on Computers*, vol. C-24, no. 12, Dec. 1975.

[6] K.E. Batcher, "The flip network in STARAN," *The Proc. of Int. Conf. on Parallel Processing*, pp. 65-71, 1976.

[7] T.Y. Feng and C.L. Wu, "On a class of multistage interconnection networks," *IEEE Trans. on Computers*, vol. C-29, Aug. 1980.

[8] H.J. Siegel and R.J. McMillen, "The multistage cube: A versatile interconnection network," *IEEE Computer*, vol. 14, Dec. 1981.

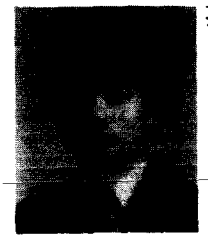
[9] H.J. Siegel, *Interconnection Networks for Large-Scale Parallel Processing: Theory and Case Studies*, Lexington Books, 1985.

[10] S.M. Reddy and V.P. Kurma, "On multipath multistage interconnection networks," *Proc. Int'l. Conf. on Dist. Computer System*, 1985.

[11] G.R. Goke and G.J. Lipvski, "Banyan networks for partitioning multiprocessor systems," *The 1st Symp. on Computer Architecture*, Dec. 1973.

[12] N.J. Davis IV and H.J. Siegel, "The performance analysis of partitioned circuit switched multistage interconnection networks," *The 12th Symp. on Computer Architecture*, June 1985. *

著 者 紹 介



孫 有 翼(正會員)
1976年 경북대학교 전자공학과 졸업. 1979年 경북대학교 대학원 전자공학과 석사학위 취득. 1987年 경북대학교 대학원 전자공학과박사과정 수료, 1979年～1984年 한국전자기술연구소 선임연구원. 1984年～현재 계명대학교 전자계산학과 조교수. 주관심분야는 컴퓨터구조, 병렬처리구조, Interconnection Network 등임.

安 光 善(正會員) 第25卷 第7號 參照
현재 경북대학교 전자계산기공학과 교수