

A Parallel Routing Algorithm on Circulant Networks Employing the Hamiltonian Circuit Latin Square

Dongkil Tak¹, Yongeun Bae¹, Chunkyun Youn² and Ilyong Chung¹

¹ Department of Computer Science, Chosun University, Kwangju, Korea
iyc@chosun.ac.kr

² Information Technology Division, Honam University, Kwangju, Korea
chqyou@itc.honam.ac.kr

Abstract. Double-loop and 2-circulant networks are widely used in the design and implementation of local area networks and parallel processing architectures. In this paper, we investigate the routing of a message on circulant networks, that is a key to the performance of this network. We would like to transmit $2k$ packets from a source node to a destination node simultaneously along paths on $G(n; \pm s_1, \pm s_2, \dots, \pm s_k)$, where the i^{th} packet will traverse along the i^{th} path ($1 \leq i \leq 2k$). In order for all packets to arrive at the destination node quickly and securely, the i^{th} path must be node-disjoint from all other paths. For construction of these paths, employing the Hamiltonian Circuit Latin Square(HCLS) we present $O(n^2)$ parallel routing algorithm on circulant networks.

1 Introduction

The intense interest in interconnection network used graph-theoretic properties for its investigations and produced various interconnection schemes. Many of these schemes have been derived to optimize important parameters such as degree, diameter, fault-tolerance, hardware cost, and the needs of particular applications. Double-loop[1] and 2-circulant networks(2-CN)[2] are widely used in the design and implementation of local area networks and parallel processing architectures. These networks are defined as follows. Let n, s_1, s_2 be positive integers such that $0 < s_1 < s_2 < n/2$. A double-loop network is a directed graph $G(n; s_1, s_2)$, where n nodes labeled with integers modulo n , and 2 links per vertex such that each node i is adjacent to the 2 other nodes $i+s_1, i+s_2$. In the undirected case, which is known as a 2-circulant network and is denoted by $G(n; \pm s_1, \pm s_2)$. It is well known that $G(n; s_1, s_2)$ and $G(n; \pm s_1, \pm s_2)$ are connected iff $\gcd(n, s_1, s_2) = 1$.

The routing of message is thus a key to the performance of such networks. There are routing algorithms using well-known methods, such as the Shortest Path Algorithm(the Forward Algorithm)[3], the Backward Algorithm[4], the Spanning

* Corresponding Author : Ilyong Chung(iyc@chosun.ac.kr)

Tree Algorithm[8]. These algorithms provide for only sequential transmission, from the source node to the desired node in a short time. We now look for algorithms that are capable of handling, multiple data items simultaneously transmitted from the starting(source) node to the destination node. There are a few algorithms on the n-dimensional hypercube network[5]-[6] that allow us to locate n disjoint paths such as the Disjoint Path Algorithm[7] and the Hamiltonian path Algorithm [8]. In this paper, we propose the algebraic approach to the routing of message on the $G(n; \pm s_1, \pm s_2, \dots, \pm s_k)$. As described above, $2k$ packets are simultaneously transmitted from the starting(source) node to the destination node. In order for all packets to arrive at the destination node quickly and securely, the i^{th} path must be node-disjoint from all other paths.

2 Design of the Hamiltonian Circuit Latin Square to the Parallel Routing Algorithm on Circulant Networks

Let A and B be any two nodes on $G(n; \pm s_1, \pm s_2, \dots, \pm s_k)$. The paper's objective is to find algorithms that will facilitate the transmission of data from node A to B in that network. In order for the data to traverse from node A to node B, it must cross, successively, intermediate nodes along a path.

Definition 1 : The routing function R for $\pm s_i$ is as follows:

$$R(A) = A \pm s_i \pmod{n}, \text{ where } A \text{ is node address}$$

Definition 2 : The relative address r of nodes A and B on $G(n; \pm s_1, \pm s_2, \dots, \pm s_k)$ is computed as the value of difference between A and B.

$$r = B - A$$

Let two addresses of node A and node B be 1 and 3. What is the relative address of two nodes? The value of the relative address is 2.

Definition 3 : Let $T(A,S)$ be the logical transmission path of data starting from node A to the destination node B, where S is a multiset and a sequence of operations, via which data can reach at the destination node. $T(A,S)$ is determined by the order of the elements in the set S. between A and B.

The i^{th} packet is transmitted along the i^{th} path, the first intermediate node of which is obtained from applying the i^{th} operation at a starting node and the last intermediate node transmits the packet to a destination node by applying the i^{th} operation. In some cases, the two operations can be the same.

Definition 4 : Let O^s be a set of operations occurring at a starting node when four packets are transmitted simultaneously and Let O^d be a set of operations occurring at a destination node when four packets arrive. These sets are defined as follows:

$$\begin{aligned}
O^s &= \{s_1, -s_1, s_2, -s_2, \dots, s_k, -s_k\} \\
O^d &= \{p_1, p_2, p_3, p_4, \dots, p_{2k-1}, p_{2k}\} \\
O^s &= O^d
\end{aligned}$$

We now apply the HCLS(Hamiltonian Circuit Latin Square) to find a set of m shortest and node-disjoint paths.

Definition 5 : The HCLS M^1 is constructed as follows: Given distinct m points $a_0, a_2, \dots, a_{m-2}, a_{m-1}$, a Hamiltonian circuit $a_i \rightarrow a_j \rightarrow \dots \rightarrow a_k \rightarrow a_i$ is randomly selected. On the circuit each row of M can be obtained from the Hamiltonian path, starting at any position $a_k(0 \leq k \leq m-1)$, under the condition that no two rows begin at the same position. If a Hamiltonian path is $a_i \rightarrow a_j \rightarrow \dots \rightarrow a_k$, then the row obtained from it is $[a_i, a_j, \dots, a_k]$

Definition 6 : Given the HCLS $M^1 = [a_{i,j}]$, the MHCM M^2 is constructed as follows: $M^2 = [A_{i,j}]$, $A_{i,j} = \{a_{i,0}, a_{i,1}, \dots, a_{i,j-1}, a_{i,j}\}$, $0 \leq i, j \leq m-1$.

We now propose a parallel routing algorithm that generates a set of m minimum-distance and node-disjoint paths for the network.

CN_Routing_Algorithm

A \leftarrow an address of a starting node A
B \leftarrow an address of a destination node B
 $O^s \leftarrow$ a set of operations occurring at a starting node A
 $O^d \leftarrow$ a set of operations requisite for reaching to a destination node B

begin

- (1) Compute the relative address R of nodes A and B; $R = B - A$
- (2) Using the relative address R, a sequence S of operations to arrive at node B in a short time are produced
- (3) In order to design a set of shortest and node-disjoint paths, find a set S_1 of distinct elements in S. A set of $|S_1|$ shortest and node-disjoint paths are generated. Each path of length is $|S|$,

- (3-1) Using the set S_1 , $(n \times n)$ HCLS is constructed, where $n = |S_1|$.
- (3-2) Operations in the i^{th} row of the HCLS are performed for traversal of the i^{th} packet and the remaining operations in S should be executed at the point except the first and the last points.
- (3-3) $O^s \leftarrow O^s - S_1$ and $O^d \leftarrow O^d - S_1$.

- (4) Construct two node-disjoint paths, each path has length $|S|+2$.

- (4-1) If $O^s = \phi$, the process is finished.
- (4-2) If a set of $\{s_i, -s_i\}$ is found in O^s , then these operations are performed at the first and the last steps of two paths newly designed, and operations in S at the middle steps of them, otherwise go to (5).
- (4-3) $O^s \leftarrow O^s - \{s_i, -s_i\}$, $O^d \leftarrow O^d - \{s_i, -s_i\}$ and go to (4-1).

(5) Generate the remaining paths.

(5-1) If $O^s = \phi$, the process is finished.

(5-2) Produce a sequence S_2 of minimum number of operations by reducing the size of $S \cup \{-s_i, -s_i\}$, $s_i \in O^s$, $S_2 = \{s_i, \min(S \cup \{-s_i, -s_i\}), s_i\}$.

(5-3) Operation g_i is performed at the first and the last steps at traversal and operations of S_2 are executed at the middle steps.

(5-4) $O^s \leftarrow O^s - \{s_i\}$, $O^d \leftarrow O^d - \{s_i\}$ and go to (5-1).

end.

3 Conclusion

In this paper, we present the algorithm that generates a set of $2k$ shortest and node-disjoint paths on $G(n; \pm s_1, \pm s_2, \dots, \pm s_k)$, employing the Hamiltonian Circuit Latin Square (HCLS). Even n and k are fixed values, the algorithm can be easily extended on arbitrary circulant networks. Important steps for determining time complexity requisite for the algorithm are two things. One is to design the HCLS, which needs $O(n)$. The other is to execute Step (5) of *CN_Routing_Algorithm*, which requires $O(n^2)$. Therefore, we can create $O(n^2)$ parallel routing algorithm for constructing $2k$ shortest and node-disjoint paths.

References

1. Bermond, J., Comellas, F., Hsu, D., "Distributed Loop Computer Networks: A Survey," J. Parallel and Distributed Computing, Academic Press, no. 24, pp.2-10, 1995.
2. Park, J., "Cycle Embedding of Faulty Recursive Circulants," J. of Korea Info. Sci. Soc., vol.31, no. 2, pp. 86-94, 2004.
3. Basse, S., Computer Algorithms : Introduction to Design and Analysis, Addison-Wesley, Reading, MA, 1978.
4. Stallings, W., Data and Computer Communications. Macmillan Publishing Company, New York, 1985.
5. Bae, M. and Bose, B., "Edge Disjoint Hamiltonian Cycles in k-ary n-cubes and Hypercubes," IEEE Trans. Comput., vol. 52, no. 10, pp. 1259-1270, 2003.
6. Thottethodi, M., Lebeck, A., and Mukherjee, S., "Exploiting Global Knowledge to Achieve Self-Tuned Congestion Control for k-ary n-cube Networks," IEEE Trans. Parallel and Distributed Systems, vol 15, no. 3, pp. 257-272, 2004.
7. Johnson, S.L. and Ho, C-T., "Optimum Broadcasting and Personalized Communication in Hypercube," IEEE Trans. Comput., vol. 38, no. 9, pp. 1249-1268, Sep. 1989.
8. Rabin, M.O., "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," J. ACM, vol. 36, no. 2, pp. 335-348, Apr. 1989.