


Article

A Novel Interconnection Network with Improved Network Cost Through Shuffle-Exchange Permutation Graph

Bo-Ok Seong¹, Hyeong-Ok Lee¹, Jong-Seok Kim² and Jung-Hyun Seo^{3,*} 

¹ Department of Computer Education, National University of Sunchon, Sunchon 57922, Korea; qjrl33@naver.com (B.-O.S.); oklee@scnu.ac.kr (H.-O.L.)

² Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707, USA; rockhee7@gmail.com

³ Department of Multimedia Engineering, National University of Chonnam, Chonnam 59626, Korea

* Correspondence: jhseo@scnu.ac.kr

Abstract: The interconnection network represents an interconnected structure of processors that strongly determines the performance quality of a parallel processing system. The shuffle-exchange permutation (SEP) network with three degrees has high fault tolerance and can be efficiently simulated through star, bubble-sort, and pancake graphs. This study proposes a new interconnection network: the new SEP (NSEP), which improves the diameter and reduces network cost by adding one edge to the SEP network, and presents its graph properties and routing algorithms. The NSEP network, with a degree of connectivity of four, demonstrated maximum fault tolerance and Hamiltonian cycle. Furthermore, the diameter was seen to be improved by 40% or more and the network cost by 20% or more.

Keywords: parallel computers; interconnection network; number of degrees; network cost; fault tolerance



Citation: Seong, B.-O.; Lee, H.-O.; Kim, J.-S.; Seo, J.-H. A Novel Interconnection Network with Improved Network Cost Through Shuffle-Exchange Permutation Graph. *Electronics* **2021**, *10*, 943. <https://doi.org/10.3390/electronics10080943>

Academic Editor: Domenico Ursino

Received: 11 March 2021

Accepted: 12 April 2021

Published: 15 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the explosive increase in data size owing to the recent advancements in information technology, the demand for high-performance computers with large computational power is increasing, particularly for big data and artificial intelligence applications. In response to these demands, high-performance computers with various computational processing units, such as graphics processing units (GPUs) and multicore processors, in addition to the conventional central processing units (CPUs), have been developed. Such computers are constantly evolving in response to new demands and requirements [1].

A parallel computer is a computer system that divides a given task and processes tasks among processing units operating in parallel. Parallel computers are classified into shared memory multiprocessors and message-passing multicomputers [2]. In the former, the memory system affects the overall system performance [3]. The interconnection network refers to the location and connection structure between processors and is one of the factors that determine the performance of a parallel processing system [4]. Hence, continuous research on interconnection networks is required to improve the performance of parallel processing computers.

Network cost is one of the measures of interconnection networks and is represented by the product of the number of degrees and the diameter. The number of degrees is related to the hardware cost and the diameter to the software cost. The network cost may be reduced by reducing the number of degrees or the diameter. The number of degrees is inversely correlated to the diameter. It is difficult to reduce the network cost because reducing the number of degrees increases the diameter, whereas reducing the diameter increases the number of degrees [4].

The interconnection networks mesh the hypercube and star graph classes depending on the number of nodes. The SEP [5] network is a star graph class with $n!$ nodes, with

node and edge symmetry, has excellent scalability through recursive structures, and has a very small number of degrees and diameters over hypercube [6–8]. The existing *SEP* network has a maximum fault tolerance with a degree of connectivity of three, and efficient simulation can be performed for star, bubble sort, and pancake graphs. Thus, one can still get the advantage of the fixed degree of the network (independent of the size) [5]. In addition, *NSEP* networks with increased degree one also predict that simulation will be efficient for star, bubble sort, and pancake graphs.

For n -dimensional *NSEP* proposed in this study, when $n = 2k$, the distance between two nodes, $\frac{n}{2}$, was reduced to one by adding an edge to the *SEP*. The proposed *NSEP* has a fixed number of degrees of four and has the properties of the existing *SEP*. The *NSEP* network has a maximum fault tolerance with a degree of connectivity of four and has a Hamilton cycle. Compared to the *SEP* network, the diameter was improved by more than 40% and the network cost by more than 20%.

In Section 2, we examine the network measure of the interconnection network, a constant-degree graph. In Section 3, we define the new interconnection network, *NSEP_n*, present the theoretical properties of graph and routing algorithm, and analyze the diameter. Finally, Section 4 concludes this study.

2. Related Works

In this chapter, we first consider the importance of the network measure and the advantages of the fixed number of degrees. Next, we examine the Hamiltonian cycle and *SEP* with an improved network cost.

In a multiprocessor system, a connected network for supporting the communication between each processor is called a multiprocessor interconnection network [9]. The interconnection network can be represented as an undirected graph representing each processor as a node and a communication link between processors as an edge. An edge is placed between any two processors with a link between them. This edge is an undirected edge that can bidirectionally transmit data. The interconnection network of parallel computers is represented as an undirected graph as follows:

$$G = (V, E)$$

where $V(G)$ is the set of nodes of graph G ; that is, $V = \{0, 1, 2, \dots, N - 1\}$ and $E(G)$ is the set of edges of graph G . The edge of graph G is a pair of arbitrary two nodes, v and w of $V(G)$. The necessary and sufficient condition for the existence of an edge (v, w) is the presence of a communication link between nodes v and w [10–15].

Network measures for evaluating interconnection networks include number of degrees, diameter, network cost, connectivity, fault tolerance, and symmetry [10,15]. The number of degrees for a node v refers to the number of edges adjacent to the node v , and the number of degrees for graph G refers to the maximum value among the number of degrees of the nodes belonging to $V(G)$. A network that has an equal number of degrees for all nodes in graph G is called a regular network. The diameter is the maximum value of the shortest path between any two nodes in the network and is the lower limit of the delay time required to transmit information to the entire network. A network having a relatively small diameter compared to the number of nodes, despite the short distance between nodes, has a disadvantage that it is difficult to design the network in terms of hardware as the number of nodes increases [16–18]. The interconnection networks that have been proposed until now can be classified into the following three types according to the number of nodes: mesh class with $k \times n$ nodes, hypercube class with 2^n nodes, and star graph class with $n!$ nodes [6].

The mesh structure has been widely used as a planar graph to date, and commercialized in various systems [19,20]. An m -dimensional mesh $M_m(N)$ consists of N^m nodes and $mN^m - mN^{m-1}$ edges. Each node's address is represented by an m -dimensional vector, and when the addresses of any two nodes differ by one in one dimension, there is an edge between them. Because low-dimensional meshes are easy to design and are useful

from the algorithmic viewpoint, they are widely used as a network of parallel processing computers. The higher the dimension of a mesh, the smaller its diameter and the larger the bisection width, and various parallel algorithms can be rapidly executed; however, it is costly [6]. Structures that improve the diameter of a mesh with a typical lattice structure, hexagonal mesh, toroidal mesh, diagonal mesh, honeycomb mesh, and torus have been proposed [19,21].

The Hamiltonian path of the interconnection network is a path that passes through all nodes of G only once. The Hamiltonian cycle of the graph G refers to a path with the same starting and destination nodes as the path that passes through all nodes only once. If the network has a Hamiltonian path or Hamiltonian cycle, a ring or a linear array can be easily implemented, which can be utilized as a useful pipeline for parallel processing [22]. If the graph v contains a Hamiltonian cycle, it is appropriate to include the Hamiltonian path.

The n -dimensional SEP graph SEP_n is a regular network that represents nodes by permutation of each symbol and has three degrees [5]. In this respect, this study interchangeably uses nodes and permutation. There are three edges of $SEP_n - \{g_{12}, g_L, g_R\}$ according to the conditions. If an arbitrary node of SEP_n is $S = s_1 s_2 s_3 \cdots s_{n-1} s_n$, adjacent nodes are as follows.

1. Edge g_{12} : Connects the nodes in which the leftmost first and the second symbols are exchanged in permutation. For example, it corresponds to a node $g_{12}(S) = s_2 s_1 s_3 \cdots s_{n-1} s_n$ that is adjacent by an edge g_{12} in a node S .
2. Edge g_L : All symbols in the permutation are moved one digit to the left, and the leftmost symbol is moved to the rightmost position. For example, it corresponds to a node $g_L(S) = s_2 s_3 \cdots s_{n-1} s_n s_1$ that is adjacent by the edge g_L in node S .
3. Edge g_R : All symbols of the node permutation are moved to the right by one digit, and the rightmost symbol is moved to the leftmost position. For example, it corresponds to a node $g_R(S) = s_n s_1 s_2 s_3 \cdots s_{n-1}$ that is adjacent by the edge g_R in node S .

In node S , any node that is adjacent by edge operation g_{12} is represented by $g_{12}(S)$, and the same method is applied to edge $\{g_L, g_R\}$. If the order of the edge sequence is $\langle g_L, g_{12}, g_R \rangle$ when the edge operation is applied in the node S , the permutation change of this node is represented as $S \rightarrow g_L(S) \rightarrow g_{12}(g_L(S)) \rightarrow g_R(g_{12}(g_L(S)))$. The permutations of the node to which the edge sequence order $\langle g_L, g_{12}, g_R \rangle$ is applied in the node S are $S = s_1 s_2 s_3 \cdots s_{n-1} s_n$, $g_L(S) = s_2 s_3 \cdots s_{n-1} s_n s_1$, $g_{12}(g_L(S)) = s_3 s_2 \cdots s_{n-1} s_n s_1$, and $g_R(g_{12}(g_L(S))) = s_1 s_3 s_2 \cdots s_{n-1} s_n$. Thus, when edge sequence $\langle g_L, g_{12}, g_R \rangle$ is applied in node S , the last node is $g_R(g_{12}(g_L(S))) = s_1 s_3 s_2 \cdots s_{n-1} s_n$. Figure 1 shows a 4D SEP_4 graph.

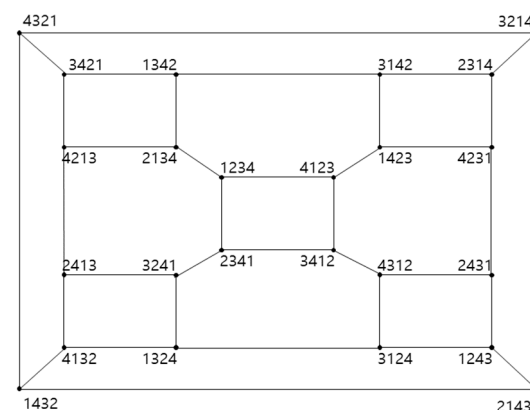


Figure 1. SEP_4 graph.

The SEP graph can be easily simulated on graphs based on permutation groups, such as a Cayley graph, and its algorithms can be efficiently executed in new graphs with minimal changes. The diameter of the SEP graph is $\frac{1}{8}(9n^2 - 22n + 24)$, and its degree of connectivity is three, having a maximum fault tolerance [5]. Because SEP_n is a Cayley graph, it has a node symmetric property [10]. The cycle, whose path length is n and

composed of edges g_L (or g_R) in SEP_n , is called s -cycle [5]. In the SEP graph, the positions of symbols are exchanged using edge operation g_{12} , and the symbol to be exchanged is moved to the leftmost position using edge operation $\langle g_L, g_R \rangle$. This study improved the diameter value by adding one edge in which the symbol of the position $\frac{n}{2} + 1$ can be quickly moved to the leftmost position in the node permutation of the graph SEP_n .

3. Definition of New Interconnection Network $NSEP_n$ and Routing Algorithm

3.1. Definition and Properties of $NSEP_n$ Graph

The $NSEP_n$ ($New - SEP_n$) graph is a regular graph with four degrees obtained by adding one edge to the existing SEP_n graph ($n = 2k, k \geq 2$). One edge added to $S = s_1 s_2 s_3 \cdots s_{\frac{n}{2}} s_{\frac{n}{2}+1} \cdots s_{n-1} s_n$, a node of $NSEP_n$, is an edge that connects the permutations in which the symbols $\frac{n}{2}, s_1 \sim s_{\frac{n}{2}}$, and $s_{\frac{n}{2}+1} \sim s_n$ have been exchanged. Let $g_{\frac{n}{2}}$ be an added edge of the $NSEP_n$ graph. The node is $g_{\frac{n}{2}}(s) = s_{\frac{n}{2}+1} \cdots s_{n-1} s_n s_1 s_2 s_3 \cdots s_{\frac{n}{2}}$ adjacent by the edge $g_{\frac{n}{2}}$ in the node $S = s_1 s_2 s_3 \cdots s_{\frac{n}{2}} s_{\frac{n}{2}+1} \cdots s_{n-1} s_n$. Therefore, the $NSEP_n$ graph has four edges $\{g_{12}, g_L, g_R, g_{\frac{n}{2}}\}$ for each node. The four nodes $\{g_{12}(S), g_L(S), g_R(S), g_{\frac{n}{2}}(S)\}$ adjacent to the node $S = s_1 s_2 s_3 \cdots s_{\frac{n}{2}} s_{\frac{n}{2}+1} \cdots s_{n-1} s_n$ of the $NSEP_n$ graph are shown below.

$$g_{12}(S) = s_2 s_1 s_3 \cdots s_{\frac{n}{2}} s_{\frac{n}{2}+1} \cdots s_{n-1} s_n$$

$$g_R(S) = s_n s_1 s_2 s_3 \cdots s_{\frac{n}{2}} s_{\frac{n}{2}+1} \cdots s_{n-1}$$

$$g_L(S) = s_2 s_3 \cdots s_{\frac{n}{2}} s_{\frac{n}{2}+1} \cdots s_{n-1} s_n s_1$$

$$g_{\frac{n}{2}}(S) = \frac{s_{\frac{n}{2}+1}}{2} \cdots s_{n-1} s_n s_1 s_2 s_3 \cdots s_{\frac{n}{2}}$$

Figure 2 shows an example of the $NSEP_4$ graph. In Figure 2, the thick line represents the edge g_{12} , the solid line represents the edge g_L (or g_R), and the dotted line represents the edge $g_{\frac{n}{2}}$ ($= g_{\frac{n}{2}}$). For example, in the case of node 1234 in the $NSEP_4$ graph, adjacent nodes are $g_{12}(1234) = 2134$, $g_L(1234) = 2341$, $g_R(1234) = 4123$, and $g_{\frac{n}{2}}(1234) = 3412$.

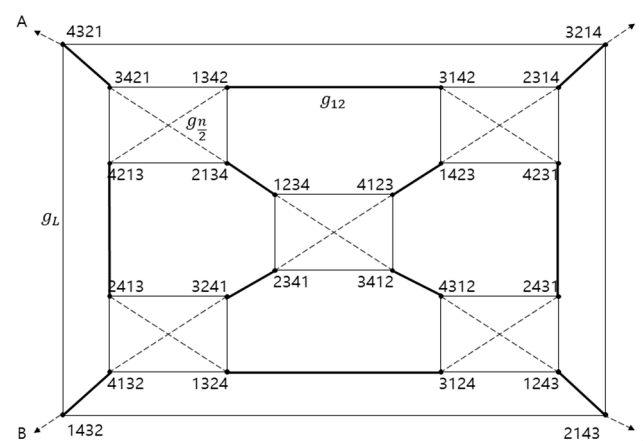


Figure 2. $NSEP_4$ graph.

Because the $NSEP_n$ graph has one extra edge over the SEP_n graph, the latter is a subgraph of the former. Cycles whose path length is n and which comprise the edges g_L (or g_R) of the $NSEP_n$ are called s -cycles. For example, an s -cycle with the path length of four at the node $S (= 1234)$ of $NSEP_4$ is

$$S = (1234) \xrightarrow{g_L} 2341 \xrightarrow{g_L} 3412 \xrightarrow{g_L} 4123 \xrightarrow{g_L} 1234 (= S)$$

A cluster in $NSEP_n$ has several important properties. These properties can be used to confirm that the $NSEP_n$ graph has a Hamiltonian cycle. The following definitions define the cluster and show its properties in Attributes 1, 2, and 3.

Definition 1. In the graph $NSEP_n$, a partial graph consisting of nodes constituting s -cycles and the edge $g_{\frac{n}{2}}$ connecting the nodes in the s -cycles is called a graph C_n .

In $NSEP_4$, one cluster C_4 containing the node $S (=1234)$ is a partial graph consisting of four edges g_L (or g_R), and two edges $g_{\frac{n}{2}}$. Figure 3 shows C_4 , the cluster of $NSEP_4$.

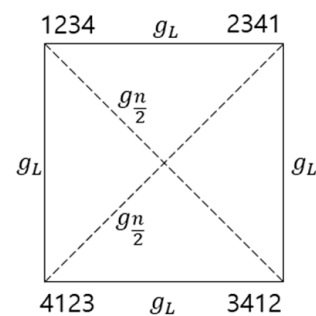


Figure 3. A cluster C_4 .

Property 1. There are $(n - 1)!$ C_n clusters in the $NSEP_n$ graph.

Proof. The total number of nodes in $NSEP_n$ is $n!$. A cluster C_n is s -cycles with n different nodes and consists of $\frac{n}{2} g_{\frac{n}{2}}$ edges that connect nodes along the path constituting s -cycles. Moreover, the number of nodes in each cluster C_n is n by s -cycles. Therefore, the number of C_n clusters is $\frac{n!}{n} = (n - 1)!$. \square

Property 2. The cluster C_n of $NSEP_n$ has $\frac{n}{2} g_{\frac{n}{2}}$ edges.

Proof. There are n nodes in each cluster in the $NSEP_n$ graph, and they are adjacent to each other by $g_{\frac{n}{2}}$ edges that exchange $\frac{n}{2}$ symbols. Because there is only one node with such an adjacent relationship for one node, two nodes form a pair. Therefore, there are $\frac{n}{2} g_{\frac{n}{2}}$ edges connecting n nodes that constitute the cluster. \square

Property 3. A node U constituting one cluster C_n of $NSEP_n$ is adjacent to the node $g_{12}(U)$ of another cluster C_n by the edge g_{12} . The n nodes constituting a cluster C_n are adjacent to nodes of n different clusters C_n by the edge g_{12} .

Proof. By the definition of $NSEP_n$ it can be seen that n nodes of a cluster C_n are adjacent to n nodes of different clusters C_n by the edge g_{12} . \square

Due to the added edge $g_{\frac{n}{2}}$, a new cycle with $(\frac{n}{2} + 1)$ nodes exists. Definition 2 defines the cycles of $NSEP_n$ and the associated theorem is shown in Lemma 1, 2, and 3. In the Lemma 4, 5, and 6, we show that there is a Hamiltonian cycle between two adjacent nodes in cluster C_n of $NSEP_n$.

Definition 2. When there is an arbitrary node U in the cluster C_n , let $V (= g_{\frac{n}{2}}(U))$ be the node U and the node adjacent to the edge $g_{\frac{n}{2}}$ (however, $U \neq V$). Let the $(\frac{n}{2} + 1)$ -cycle be the path from node U to the node V constituting the edge g_L (or g_R) at the path distance of $\frac{n}{2}$, and the path constituting the edge $g_{\frac{n}{2}}$ at the node U .

Assume that there are nodes $U(= 1234)$ and $V(= 3412)$ at $NSEP_4$. The three-cycle path containing node $U(= 1234)$ and $V(= 3412)$ is given as $U(= 1234) \xrightarrow{g_R} 4123 \xrightarrow{g_R} 3412(= V) \xrightarrow{g_L} 1234(= U)$.

Lemma 1. When there is a node U in one cluster C_n , let $V(= g_{\frac{n}{2}}(U))$ be the node U and the node adjacent to the edge $g_{\frac{n}{2}}$ (however, $U \neq V$). There are two $(\frac{n}{2} + 1)$ -cycles that share the edge $g_{\frac{n}{2}}$ connecting nodes U and V .

Proof. By Definition 2, $(\frac{n}{2} + 1)$ -cycle is a path consisting of $g_{\frac{n}{2}}$ edge g_R (or g_L), and 1 $g_{\frac{n}{2}}$ edge. It can be seen that these $(\frac{n}{2} + 1)$ -cycles can create cycles using the edge g_R and the edge g_L , respectively. Therefore, there are two $(\frac{n}{2} + 1)$ -cycles that share the edge $g_{\frac{n}{2}}$. \square

Lemma 2. In each cluster of $NSEP_n$, there are $n(\frac{n}{2} + 1)$ -cycles.

Proof. By Property 2, each cluster has $\frac{n}{2} g_{\frac{n}{2}}$ edges, and by Lemma 1, there are two $(\frac{n}{2} + 1)$ -cycles that share an edge $g_{\frac{n}{2}}$. Therefore, because $\frac{n}{2} \times 2 = n$, there are $n(\frac{n}{2} + 1)$ -cycles. \square

Lemma 3. The number of $(\frac{n}{2} + 1)$ -cycles in the network $NSEP_n$ is $n!$.

Proof. By Lemma 2, there are $n(\frac{n}{2} + 1)$ -cycles in each cluster, and by Property 1, the number of C_n clusters is $(n - 1)!$. Therefore, the number of $(\frac{n}{2} + 1)$ -cycles in the $NSEP_n$ network is $(n - 1)! \times n = n!$. \square

Lemma 4. There is a Hamiltonian path, whose path length is n , including an arbitrary node U in the cluster C_n , and a node $g_R(U)$ (or $g_L(U)$) adjacent to the edge g_R (or g_L) from the node U .

Proof. Let U be an arbitrary node of the cluster C_n . Let $V1(= g_L(U))$ be the adjacent node by node U and edge g_L , and the node adjacent by node and edge. Because each cluster C_n has s -cycles of $NSEP_n$ as a partial graph, the path connected by node U and edge g_R (or g_L) has cycles including nodes $V1$ and $V2$. Therefore, there is a Hamiltonian cycle with the path length of n from node U to an adjacent node $V1(= g_L(U))$ by an edge g_L , and an adjacent node $V2(= g_R(U))$ by the node U and the edge g_R . \square

Lemma 5. There is a Hamiltonian cycle between an arbitrary node U that constitutes a cluster C_n , and nodes $V1(= g_{\frac{n}{2}}(U))$ connected from the node U to the edge $g_{\frac{n}{2}}$.

Proof. Let U be the starting node of the cluster C_n , and the target node $V1(= g_{\frac{n}{2}}(U))$ be the node that is connected to node U and the edge $g_{\frac{n}{2}}$. By the definition of $NSEP_n$ graph, the distance between the nodes U and V in s -cycles consisting of edges g_L (or g_R) is $\frac{n}{2}$. Let $S1$ be a node at a $(\frac{n}{2} - 1)$ distance along the s -cycle from the starting node U . The node $S2(= g_{\frac{n}{2}}(S1))$ connected by the node $S2$, and the edge $g_{\frac{n}{2}}$ has a distance of $\frac{n}{2}$ in s -cycles. Therefore, node $S2$ is a node adjacent to U located at a distance of $(\frac{n}{2} - 1)$ from the node $S1$. A node at a $(\frac{n}{2} - 1)$ distance along the s -cycle from a node $S2$ becomes a target node V . Because nodes U and $V1(= g_{\frac{n}{2}}(U))$ are adjacent to the edge $g_{\frac{n}{2}}$, a Hamiltonian cycle is formed. Therefore, there is a Hamiltonian cycle with a length n , connecting two adjacent nodes in the cluster C_n . \square

Lemma 6. There exists a Hamiltonian cycle that includes two adjacent nodes U, V in the cluster C_n .

Proof. By Lemmas 4 and 5, there is a Hamiltonian cycle that includes two adjacent nodes U, V in the cluster C_n . \square

The reduced graph RS_{n-1} of $NSEP_n$ represents the reduced s-cycles in $NSEP_n$ to one node. The node whose leftmost symbol is one in the permutation of n nodes constituting the s-cycles of $NSEP_n$ is called the leader node. The node address of the reduced graph RS_{n-1} is represented by the remaining permutation addresses except one in the permutation of the leader node. In s-cycles 1234–4123–3412–2341, shown in Figure 4, the leader node is 1234, and s-cycle is represented by the super node 234 in the graph RS_3 . Definition 3 defines a subgraph of RS_{n-1} as RS_{n-2}^k relative to the rightmost symbol. The theorem about it is shown in auxiliary Lemmas 7–9, and $NSEP_n$ shows that in Theorem 1 has a Hamilton cycle.

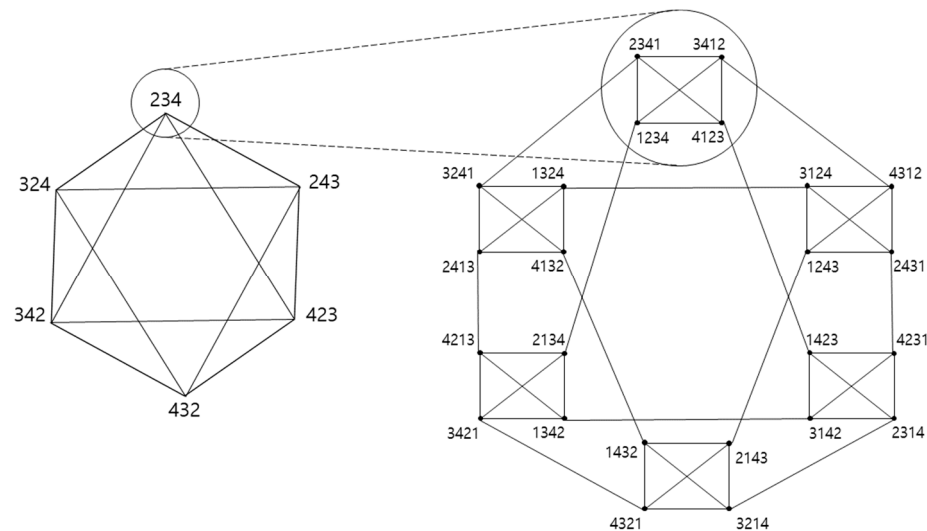


Figure 4. RS_3 and $NSEP_4$ graphs.

Definition 3. A bubble-sort graph, which is a partial graph that includes all nodes of the reduced graph RS_{n-1} , is known as RSB_{n-1} . Furthermore, RSB_{n-1} is a $(n-2)$ normal Cayley graph [5]. Therefore, when RS_{n-1} is $n \geq 5$, it includes $n-1$ subgraph RS_{n-2} , and all RS_{n-2} are adjacent to each other. Because the nodes belonging to RS_{n-2} have the same rightmost symbol, when the rightmost symbol is k , the RS_{n-2} of RS_{n-1} is defined as RS_{n-2}^k ($2 \leq k \leq n$).

The $NSEP_n$ network has an even number of $n = 2k$ symbols representing node addresses. In a network having an even number of symbols in $NSEP_n$, RS_4 does not exist. If RS_{n-1} has a Hamiltonian cycle, it is natural that there is a Hamiltonian cycle when n is even. After showing that there is a Hamiltonian cycle in RS_{n-1} , we show that there is also a Hamiltonian cycle in $NSEP_n$.

Lemma 7. There is a Hamiltonian path between any two arbitrary nodes of RS_3 , and it has a Hamiltonian cycle.

Proof. Let U and V be the starting and destination nodes, respectively. RS_3 can be divided into two areas A and B with the same number of nodes. The thick lines correspond to the edges within each area. There are three nodes constituting one area, and all nodes are adjacent. In addition, the nodes constituting the area have cycles in a complete graph, and there is always a Hamiltonian path between two nodes. In each node, there are two edges connecting to nodes in other areas. There are two cases of the relationship with nodes U and V , as shown in Figure 5. The edges can be present in one area as shown in Figure 5-1, or in different areas as shown in Figure 5-2. Let $U' (\neq V)$ and V' be a node adjacent to U in area A and a node adjacent to V in area B, respectively. There is a node adjacent to U' in area B, and there is a Hamiltonian path between this node and V' . Therefore, in Case 1, there is a Hamiltonian path between U and V . Now we move on to Case 2. Let $U' (\neq V')$

be the node connected through the Hamiltonian path from U in area A. Because both U' and V are present in area B, there is a Hamiltonian path. That is, there is a Hamiltonian path between U and V in Case 2 as well. Therefore, because there is a Hamiltonian path between any two nodes of RS_3 , and another Hamiltonian path between adjacent nodes, this has a Hamiltonian cycle. \square

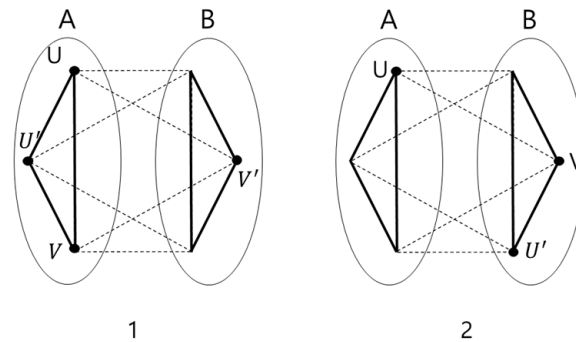


Figure 5. Areas A and B of RS_3 .

Lemma 8. If CN_{n-1} ($n \geq 5$) is the number of RS_3 in RS_{n-1} , $CN_{n-1} = \frac{(n-1)!}{3!}$ is established.

Proof. By Definition 3, RS_{n-1} has $n-1$ RS_{n-2} subgraphs as a cluster. Figure 6 shows a subgraph on the RS_{n-1} . For example, if $n = 5$, then CN_4 , which is the number of RS_3 in RS_4 becomes $CN_4 = CN_3 \times 4$. Because RS_{n-1} is hierarchical, let us assume $CN_{n-1} = CN_{n-2} \times (n-1)$. We prove that the formula $CN_{n-1} = \frac{(n-1)!}{3!}$ holds by mathematical induction. \square

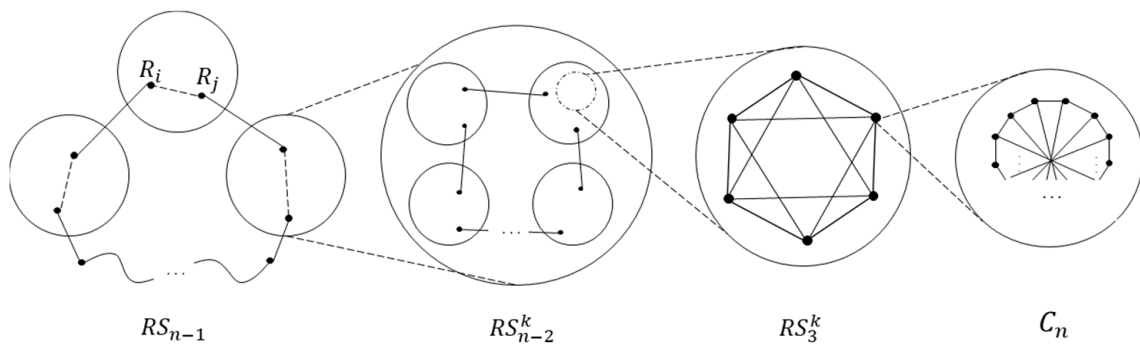


Figure 6. Relationship between RS_{n-1} graph and subgraphs.

Because the number of RS_3 in RS_3 is 1, $CN_3 = 1$.

(i) When $n = 5$, $CN_{n-1} = CN_{n-2} \times (n-1) = \frac{(n-1)!}{3!}$ holds.

This is true because $CN_4 = CN_3 \times 4 = 4$ and $CN_{n-1} = \frac{(n-1)!}{3!} = \frac{4!}{3!} = 4$.

(ii) When $n = i$, it is assumed that $CN_{i-1} = \frac{(i-1)!}{3!}$ holds.

(iii) When $n = i + 1$, we prove that it holds.

$$CN_{(i+1)-1} = CN_i = CN_{i-1} \times i = \frac{(i-1)!}{3!} \times i = \frac{(i-1)! \times i}{3!} = \frac{i!}{3!} = \frac{((i+1)-1)!}{3!}$$

Therefore, the number of RS_3 in RS_{n-1} satisfies the following equation.

$$CN_{n-1} = CN_{n-2} \times (n-1) = \frac{(n-1)!}{3!}$$

Lemma 9. RS_{n-1} has a Hamiltonian cycle.

Proof. By Definition 3, all $n - 1$ RS_{n-2}^k subgraphs are adjacent to each other in RS_{n-1} . Let $R_i (1 \leq i \leq (n - 1)!) and $R_j (1 \leq j \leq (n - 1)!, i \neq j)$ be any two nodes adjacent to each other in RS_{n-1} . Adjacent relationships are indicated by dotted lines. All nodes in RS_{n-1} are adjacent to RS_{n-2}^k through adjacent nodes. There is always adjacent RS_{n-3}^k in RS_{n-2} . Therefore, it can be seen that RS_3^k is always adjacent hierarchically in the same manner up to RS_4 . As we have shown in Lemma 7, that there exists a Hamiltonian path between any two nodes, which implies that there exists a Hamiltonian cycle in RS_{n-1} . $\square$$

Theorem 1. The $NSEP_n$ network has a Hamiltonian cycle.

Proof. By Lemma 9, RS_{n-1} has a Hamiltonian cycle, which regards the cluster C_n of $NSEP_n$ as a super node. A node in an adjacent cluster of $NSEP_n$ is adjacent to another cluster through an adjacent node [5]. By Lemma 6, there is a Hamiltonian path between adjacent nodes of the cluster C_n . Thus, the $NSEP_n$ network has a Hamiltonian cycle. Figure 7 shows a Hamiltonian cycle of RS_4 . \square

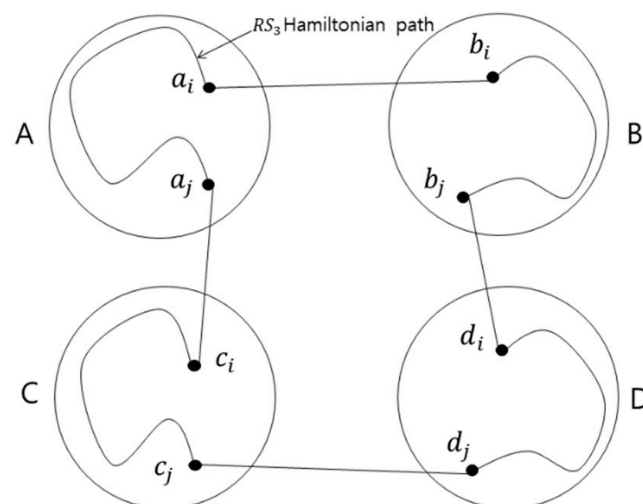


Figure 7. Hamiltonian cycle of RS_4 .

3.2. Routing Algorithm and Diameter Analysis

Routing refers to the path from one node to another. Because SEP_n a partial graph of $NSEP_n$ is a Cayley graph, it is node symmetric [10]. Therefore, the path of the starting node S and the destination node D can be regarded as the path of the starting node S and the ID node. Let the ID node be $123 \cdots n$. The algorithm proposed in this study is a method of placing the symbols in sequence up to n by iteratively applying the method of checking the positions of symbols 1 and 2, placing symbol 2 on the right side of symbol 1, and symbol 3 on the right side of symbol 2. The position of the symbol is represented in Definition 4, and the formulas used by the algorithm are represented in Lemmas 10–13.

Definition 4. The position of the symbol s_1 in the current node $S (= s_1 s_2 s_3 \cdots s_i \cdots s_{n-1} s_n)$ is represented by $p(s_i)$ ($1 \leq i \leq n$).

In the case where the node address is 256,314 in $NSEP_6$, $p(1) = 5$ and $p(2) = 1$.

Lemma 10. In node $S (= s_1 s_2 s_3 \cdots s_i \cdots s_{n-1} s_n)$, the path of the node adjacent to the node by the edge sequence $\langle g_L, g_{12}, g_R \rangle$ is as follows. The last node permutation is $g_R(g_{12}(g_L(S))) =$

$s_1s_3s_2 \cdots s_{n-1}s_n$ in the path to which the edge sequence $\langle g_L, g_{12}, g_R \rangle$ is applied in node S . The path in node S is as given below.

$$S(= s_1s_2s_3 \cdots s_i \cdots s_{n-1}s_n) \rightarrow g_L(S) = s_2s_3 \cdots s_i \cdots s_{n-1}s_ns_1 \rightarrow \\ g_{12}(g_L(S)) = s_3s_2 \cdots s_i \cdots s_{n-1}s_ns_1 \rightarrow g_R(g_{12}(g_L(S))) = s_1s_3s_2 \cdots s_i \cdots s_{n-1}s_n$$

Lemma 11. The number of iterations for an edge or edge sequence is denoted by $' \times [i]'$. For example, when $i = 3$, $\langle g_A, g_B \rangle \times [3] = \langle g_A, g_B \rangle, \langle g_A, g_B \rangle, \langle g_A, g_B \rangle$. The number of iterations of the edge sequence is incorporated as given below.

$$\langle g_A, g_B \rangle \times [i]$$

When $p(i) = a$, $p(i+1) = b$,

$$\langle g_A, g_B \rangle \times [p(i) + p(i+1) - k] = \langle g_A, g_B \rangle \times [a + b - k]$$

Lemma 12. The value of the number of iterations of the edge sequence, which is less than 0, is subject to the reverse operation.

$$\langle g_A \rangle \times [-n] = \langle g_A^{-1} \rangle \times [n]$$

Lemma 13. The distance between the symbols s_i and s_j at the node address $S = s_1s_2s_3 \cdots s_i \cdots s_j \cdots s_n$ ($n = 2^k$, $1 \leq k \leq \log_2 n$) is denoted by $|p(s_i) - p(s_j)|$.

The routing algorithm is outlined as follows.

[STEP 1] Symbol 2 is placed to the right of symbol 1. When the node address is divided by half, that is, $\frac{n}{2}$, the positions of the two symbols, $p(1)$ and $p(2)$ are checked, and the algorithm is executed according to the following cases. The cases are divided into the cases of $p(1), p(2) \leq \frac{n}{2}$; $p(1), p(2) \geq \frac{n}{2}$; $p(1) \leq \frac{n}{2}$ and $p(2) > \frac{n}{2}$, or $p(2) \leq \frac{n}{2}$ and $p(1) > \frac{n}{2}$.

[STEP 2] $i+1$ is placed to the right of symbol i . When the node address is divided by half, the positions of the two symbols, $p(i)$ and $p(i+1)$ are checked, and the algorithm is executed according to the following cases. The cases are divided into the cases of $p(i), p(i+1) \leq \frac{n}{2}$, $p(i), p(i+1) > \frac{n}{2}$, and $(p(i) \leq \frac{n}{2} \text{ AND } p(i+1) > \frac{n}{2}) \text{ OR } (p(i+1) \leq \frac{n}{2} \text{ AND } p(i) > \frac{n}{2})$.

[STEP 3] In this algorithm, n is placed at the rightmost position while the relative positions from 1 to n are arranged in an ascending order, and this is the step of matching with the target node ID .

The routing shown in Algorithm 1.

Algorithm 1. Routing**[STEP 1]**

```

1: case  $p(1), p(2) \leq \frac{n}{2}$ 
2:   if  $(p(2) - p(1) = 1)$  then goto [STEP 2];
3:   if  $(p(1) \neq 1 \text{ AND } p(2) \neq 1)$  then exit this case block
4:   if  $(k = 2) \langle g_{12} \rangle$ ;
5:   else {
6:     if  $|p(2) - p(1)| = 1 \langle g_{12} \rangle$ ;
7:     else  $\{ \langle g_{12} \rangle, \langle g_L, g_{12} \rangle \times [|p(2) - p(1)| - 2];$ 
8:     if  $(p(2) < p(1)) \langle g_L, g_{12} \rangle$ ;
9:   }
10: }
11:  $\langle g_L, g_{12} \rangle \times [|p(1) - p(2)| - 1];$ 
12: If  $(p(2) < p(1)) \langle g_L, g_{12} \rangle$ ;
13: case  $p(1), p(2) > \frac{n}{2}$ 
14:    $\langle g_{\frac{n}{2}} \rangle$ 
15:   goto case  $p(1), p(2) \leq \frac{n}{2}$ 
16:   case  $( (p(1) \leq \frac{n}{2} \text{ AND } p(2) > \frac{n}{2}) \text{ OR } (p(2) \leq \frac{n}{2} \text{ AND } p(1) > \frac{n}{2}) )$ 
17:   Described in Appendix A.

```

[STEP 2]

```

1: for  $1 < i < n - 1, (i = 2, 3, \dots, n - 3, n - 2) \{$ 
2:   case  $p(i), p(i + 1) \leq \frac{n}{2}$ 
3:     if  $(p(i + 1) - p(i) = 1 \text{ OR } 1 - n) i++;$ 
4:     elseif  $(p(i) < p(i + 1)) \{$ 
5:        $\langle g_L \rangle \times [p(i + 1) - 2], \langle g_{12} \rangle, \langle g_R, g_{12} \rangle \times [p(i + 1) - p(i) - 2]; \}$ 
6:     elseif  $(p(i) > p(i + 1)) \{$ 
7:        $\langle g_{12}, g_L \rangle \times [p(i) - p(i + 1) - 1], \langle g_{12} \rangle; \}$ 
8:     case  $p(i), p(i + 1) > \frac{n}{2}$ 
9:        $\langle g_{12} \rangle$ 
10:      goto case  $p(i), p(i + 1) \leq \frac{n}{2}$ 
11:      case  $( (p(i) \leq \frac{n}{2} \text{ AND } p(i + 1) > \frac{n}{2}) \text{ OR } (p(i + 1) \leq \frac{n}{2} \text{ AND } p(i) > \frac{n}{2}) )$ 
12:        Described in Appendix B.
13: }

```

[STEP 3]

```

1: If  $(p(n) < \frac{n}{2} - 1) \langle g_L \rangle \times [p(n);$ 
2: if  $(p(n) = \frac{n}{2} - 1) \{ \langle g_R \rangle, \langle g_{\frac{n}{2}} \rangle \};$ 
3: if  $(p(n) = \frac{n}{2}) \langle g_{\frac{n}{2}} \rangle$ ;
4: if  $(p(n) = \frac{n}{2} + 1) \{ \langle g_L \rangle, \langle g_{\frac{n}{2}} \rangle \};$ 
5: else  $(p(n) > \frac{n}{2} + 1) \langle g_R \rangle \times [n - p(n)];$ 

```

When the Algorithm 1 proposed in this study is applied, the process of sorting from nodes 2143 to 1234 is described as given below when $n = 4$.

```

[STEP 1] 2143  $\xrightarrow{g_{12}}$  1243
[STEP 2] 1243  $\xrightarrow{g_{\frac{4}{2}}}$  4312  $\xrightarrow{g_{12}}$  3412
[STEP 3] 3412  $\xrightarrow{g_{\frac{4}{2}}}$  1234

```

Theorem 2. The diameter of $NSEP_n$ is $\frac{2}{3}n^2 - \frac{3}{2}n + 1$ ($n = 2k, k \geq 2$).

Proof. In the worst case in [STEP 1], the diameter is n .

Case $((p(1) \leq \frac{n}{2} \text{ AND } p(2) > \frac{n}{2}) \text{ OR } (p(2) \leq \frac{n}{2} \text{ AND } p(1) > \frac{n}{2}))$
 $|p(1) - p(2)| \leq \frac{n}{2}$, and it satisfies the following.
 $p(n) \langle \min(p(1), p(2)) \text{ OR } p(n) \rangle \max(p(1), p(2))$

□

In Table 1, the network cost of $NSEP_n$ was compared with the constant branching class connections. $NSEP_n$ increases the number of nodes rapidly as n increases. Thus, some of the network costs for each network were rearranged in Table 2 when the number of nodes was equal, and the results were shown in Table 3 and Figure 8 as a graph. Here, the network cost of $NSEP_n$ is always less than that of SEP_n , and we can see that when it is $n > 10$, the network cost of $NSEP_n$ is the smallest.

Table 1. Network cost comparison of $NSEP$ with other fixed degree networks.

Network	Node Number	Degree	Diameter	Network Cost
Mesh	$k \times n$	4	$2\sqrt{n}$	$O(8\sqrt{n})$
Honeycomb mesh	$6n^2$	3	$1.63\sqrt{n}$	$O(4.9\sqrt{n})$
Torus	$k \times n$	4	\sqrt{n}	$O(4\sqrt{n})$
SEP	$n!$	3	$\frac{1}{8}(9n^2 - 22n + 24)$	$O\left(\frac{27}{8}n^2\right)$
$NSEP$	$n!$	4	$\frac{2}{3}n^2 - \frac{3}{2}n + 1$	$O\left(\frac{8}{3}n^2\right)$

Table 2. Network cost comparison between constant degrees class and $NSEP$.

n	Network Costs					Number of Nodes			
	Mesh	Honeycomb	Torus	SEP	$NSEP$	Mesh	Honeycomb	Torus	SEP ($NSEP$)
4	16.00	9.78	8.00	30	22.67	16	96	16	24
5	17.89	10.93	8.94	52	40.67	25	150	25	120
6	19.60	11.98	9.80	81	64.00	36	216	36	720
7	21.17	12.94	10.58	117	92.67	49	294	49	5040
8	22.63	13.83	11.31	159	126.67	64	384	64	40,320
9	24.00	14.67	12.00	208	166.00	81	486	81	362,880
10	25.30	15.46	12.65	264	210.67	100	600	100	3,628,800
11	26.53	16.22	13.27	327	260.67	121	726	121	39,916,800
12	27.71	16.94	13.86	396	316.00	144	864	144	479,001,600
13	28.84	17.63	14.42	472	376.67	169	1014	169	6,227,020,800
14	29.93	18.30	14.97	555	442.67	196	1176	196	87,178,291,200
15	30.98	18.94	15.49	645	514.00	225	1350	225	1.30767×10^{12}

Table 3. Comparison of network costs when number of nodes are equal.

Number of Nodes	Network Costs				
	Mesh	Honeycomb	Torus	SEP	$NSEP$
24	17		8.94	30	22.67
120	26.53	10	13.2	52	40.67
720	14.5	16	20.7	81	64
5040	67.4	26.3	33.7	117	92.67
40,320	113.4	44.2	56.7	159	126.67
362,880	196.45	120.08	98.22	208	166
3,628,800	349.17	213.43	174.59	264	210.67
39,916,800	635.89	388.69	317.94	327	260.67

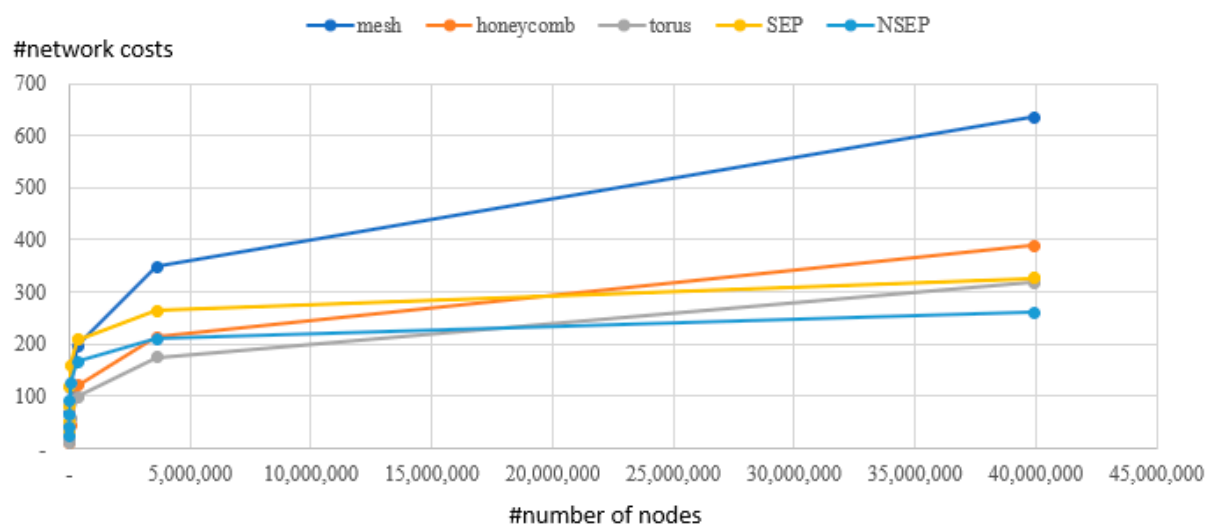


Figure 8. Network cost comparison when the number of nodes is the same.

In Figure 8, the five circles at the right end of the chart represent network costs of the mesh, honeycomb, SEP, torus, and NSEP, in that order from the top, when the number of nodes is 4×10^8 .

4. Conclusions

The SEP interconnection network has three degrees and a diameter of $\frac{1}{8}(9n^2 - 22n + 24)$. This study proposed a new interconnection network NSEP by adding a new edge to the SEP network. In the NSEP network, the diameter and network cost were improved by reducing the distance between two nodes in a $\frac{n}{2}$ distance to one by adding one edge to the existing SEP network.

The $NSEP_n$ interconnection network proposed in this study has the same number of nodes as SEP, having four degrees, a diameter of $\frac{2}{3}n^2 - \frac{3}{2}n + 1$, and a network cost of $O(n^2)$. The interconnection network $NSEP_n$ shows excellent results by reducing the diameter by 40% or more and the network cost by 20% or more, while increasing the number of degrees by one in comparison to SEP. The interconnection network NSEP is a network with a Hamiltonian cycle and SEP as a subgraph. Because the NSEP network is defined to only have an even number of nodes ($n = 2k$), a generalized graph definition is additionally required. The algorithm designed in this paper is an algorithm that sorts symbols 1 through n . In some cases, the opposite arrangement of n through 1 may be effective. Further research will be required under conditions that allow us to select efficient algorithms between the two algorithms. It is hoped that this will lead to research on interconnected networks to improve the performance of parallel processing computers.

Author Contributions: Conceptualization, B.-O.S., J.-H.S. and H.-O.L.; methodology, B.-O.S. and J.-H.S.; software, J.-H.S.; validation, J.-S.K.; formal analysis, B.-O.S. and J.-H.S.; investigation, B.-O.S. and J.-S.K.; resources, B.-O.S. and J.-H.S.; data curation, B.-O.S. and J.-S.K.; writing—original draft preparation, B.-O.S. and J.-H.S.; writing—review and editing, B.-O.S. and J.-H.S.; visualization, B.-O.S. and J.-S.K.; supervision, H.-O.L.; project administration, H.-O.L.; funding acquisition, H.-O.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1A2C1012363).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

```

1: case (  $(p(1) \leq \frac{n}{2} \text{ AND } p(2) > \frac{n}{2}) \text{ OR } (p(2) \leq \frac{n}{2} \text{ AND } p(1) > \frac{n}{2})$  )
2:   if  $(p(2) - p(1) = 1 \text{ OR } 1 - n)$  then goto [STEP 2];
3:   elseif  $(p(1) - p(2) = 1)$  then  $\langle g_R, g_{\frac{n}{2}}, g_{12} \rangle$ ;
4:   elseif  $(|p(1) - p(2)| \leq \frac{n}{2})$  {
5:     if  $(\min(p(1), p(2)) < p(n) < \max(p(1), p(2)))$  {
6:        $\langle g_L \rangle \times [\min(p(1), p(2)) - 2]$ ,
7:        $\langle g_L, g_{12} \rangle \times [|p(1) - p(2)| - 1]$ ;
8:       if  $(p(2) < p(1))$   $\langle g_L, g_{12} \rangle$ ; }
9:     else  $(p(n) \langle \min(p(1), p(2)) \text{ OR } p(n) \rangle \max(p(1), p(2)))$  {
10:       $\langle g_{\frac{n}{2}} \rangle$ ,  $\langle g_L \rangle \times [\max(p(1), p(2)) - \frac{n}{2} - 2]$ ,  $\langle g_{12} \rangle$ ;
11:      if  $(|p(1) - p(2)| \geq 2)$  {
12:         $\langle g_R, g_{12} \rangle \times [|p(1) - p(2)| - 2]$ ; }
13:      if  $(p(1) = 1)$   $\langle g_R, g_{12} \rangle$ ; }
14:   } }
15:   elseif  $(|p(1) - p(2)| > \frac{n}{2})$  {
16:     if  $(\min(p(1), p(2)) < p(n) < \max(p(1), p(2)))$  {
17:        $\langle g_L \rangle \times [\min(p(1), p(2)) - 2]$ ,
18:        $\langle g_{12}, g_R \rangle \times [n - |p(1) - p(2)| - 1]$ ; }
19:     if  $(p(2) < p(1))$   $\langle g_{12} \rangle$ ; }
20:     else  $(p(n) \langle \min(p(1), p(2)) \text{ OR } p(n) \rangle \max(p(1), p(2)))$  {
21:        $\langle g_R \rangle \times [n - \max(p(1), p(2)) + 1]$ ,
22:        $\langle g_{12}, g_L \rangle \times [n - |p(1) - p(2)| - 1]$ ;
23:       if  $(p(2) < p(1))$   $\langle g_{12} \rangle$ ; }
24:   }

```

Appendix B

```

1: case (  $(p(i) \leq \frac{n}{2} \text{ AND } p(i+1) > \frac{n}{2}) \text{ OR } (p(i+1) \leq \frac{n}{2} \text{ AND } p(i) > \frac{n}{2})$  )
2:   if  $(p(i+1) - p(i) = 1 \text{ OR } 1 - n)$  then  $i++$ ;
3:   elseif  $(|p(i+1) - p(i)| \leq \frac{n}{2})$  {
4:     if  $p(i) < p(i+1)$  {
5:        $\langle g_{\frac{n}{2}} \rangle$ ,  $\langle g_L \rangle \times [p(i+1) - \frac{n}{2} - 2]$ ,
6:        $\langle g_{12} \rangle$ ,  $\langle g_R, g_{12} \rangle \times [p(i+1) - p(i) - 2]$ ; }
7:     elseif  $(p(i) > p(i+1))$  {
8:        $\langle g_L \rangle \times [p(i+1) - 1]$ ,  $\langle g_{12} \rangle$ ,  $\langle g_L, g_{12} \rangle \times [p(i) - p(i+1) - 1]$ ; }
9:   } }
10:   elseif  $(|p(i+1) - p(i)| > \frac{n}{2})$  {
11:     if  $(p(i) < p(i+1))$  {
12:        $\langle g_R \rangle \times [n - p(i+1) + 1]$ ,
13:        $\langle g_{12} \rangle$ ,  $\langle g_L, g_{12} \rangle \times [n - p(i+1) + p(i) - 1]$ ; }
14:     elseif  $(p(i) > p(i+1))$  {
15:        $\langle g_L \rangle \times [p(i+1) - 2]$ ,  $\langle g_{12} \rangle$ ,
16:        $\langle g_{12} \rangle$ ,  $\langle g_R, g_{12} \rangle \times [n - p(i) + p(i+1) - 2]$ ; }
17:     if  $(p(2) < p(1))$   $\langle g_{12} \rangle$ ; }
18:   }

```

References

- Kim, N.-G. *A Study on the Development of the Composite Measures of High-Performance Computer Technology*; Sungkyunkwan University: Seoul, Korea, 2020.
- Dadheech, P.; Kumar, A. Fault-Tolerant Adaptive XY Routing for Multiprocessors in HPC Network. *J. High Perform. Comput.* **2020**, *3*, 94–118. [\[CrossRef\]](#)
- Culler, D.; Singh, J.P.; Gupta, A. *Parallel Computer Architecture—A Hardware/Software Approach*; Gulf Professional Publishing: Houston, TX, USA, 1999.
- Steller, P. A survey of the Degree/Diameter Problem for Undirected Graphs. Ph.D. Thesis, University of Delaware, Newark, NJ, USA, 2020; pp. 1–5.

5. Latifi, S.; Srimani, P.K. *A New Fixed Degree Regular Network for Parallel Processing*; IEEE Computer Society Press: New York, NY, USA, 1996; pp. 152–159.
6. Adhikari, N.; Singh, A. Leafycube: A Novel Hypercube Derivative for Parallel Systems. In *Advances in Data Science and Management*; Springer: Singapore, 2020; pp. 323–332.
7. Gu, M.M.; Hao, R.X.; Tang, S.M.; Chang, J.M. Analysis on component connectivity of bubble-sort star graphs and burnt pancake graphs. *Discret. Appl. Math.* **2020**, *279*, 80–91. [[CrossRef](#)]
8. Yeh, C.-H.; Varvarigos, E. Macro-Star Network: Efficient Low-Degree Alternatives to Star Graphs for Large-Scale Parallel Architectures. *IEEE Trans. Parallel Distrib. Syst.* **1996**, *9*, 987–1003.
9. Gholizadeh, R.; Valinataj, M. Reliability Improvement of Fault-Tolerant Shuffle Exchange Interconnection Networks. In Proceedings of the 2020 10th International Conference on Computer and Knowledge Engineering (ICCCKE), Mashhad, Iran, 29–30 October 2020; pp. 336–341.
10. Akers, S.B.; Krishnamurthy, B. A group-theoretic model for symmetric interconnection networks. *IEEE Trans. Comput.* **2018**, *38*, 555–566. [[CrossRef](#)]
11. Mnejja, S.; Aydi, Y.; Abid, M.; Monteleone, S.; Catania, V.; Palesi, M.; Patti, D. Delta multi-stage interconnection networks for scalable wireless on-chip communication. *Electronics* **2020**, *9*, 913. [[CrossRef](#)]
12. Wu, H.I.; Tsay, R.S.; Chang, F.Y. CORONA: A k-CONnected ROBust Interconnection Network Generation Algorithm. In Proceedings of the 2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 10–13 August 2020.
13. Mnejja, S.; Aydi, Y.; Abid, M.; Monteleone, S.; Catania, V.; Palesi, M.; Patti, D. Hierarchical and reconfigurable optical/electrical interconnection network for high-performance computing. *IEEE/OSA J. Opt. Commun. Netw.* **2020**, *12*, 50–61.
14. Pai, K.-J. A Parallel Algorithm for Constructing Two Edge-Disjoint Hamiltonian Cycles in Crossed Cubes. In Proceedings of the International Conference on Algorithmic Applications in Management, Jinhua, China, 10–12 August 2020.
15. Lakshmivarahan, S.; Jwo, J.S.; Dhall, S.K. Symmetry in interconnection networks based on cayley graphs of permutation groups: A survey. *Parallel Comput.* **1993**, *19*, 361–407. [[CrossRef](#)]
16. Yasudo, R.; Nakano, K.; Koibuchi, M.; Matsutani, H.; Amano, H. Designing low-diameter interconnection networks with multi-ported host-switch graphs. *Concurr. Comput. Pract. Exp.* **2020**, e6115. [[CrossRef](#)]
17. Moudi, M.; Othman, M. A survey on emerging issues in interconnection networks. *Int. J. Internet Technol. Secur. Trans.* **2021**, *11*, 131. [[CrossRef](#)]
18. Mahafzah, B.A.; Alshraideh, M.; Tahat, L.; Almasri, N. Topological Properties Assessment for Hyper Hexa-Cell Interconnection Network. *Int. J. Comput.* **2019**, *13*, 115–121.
19. Bokka, V.H.; Gurla, S.; Olariu, J.L. Podality-based time-optimal computations on enhanced meshes. *IEEE Trans. Parallel Distrib. Syst.* **1997**, *8*, 1019–1035. [[CrossRef](#)]
20. Ranade, A.G.; Johnsson, S.L. *The Communication Efficiency of Meshes, Boolean Cubes and Cube Connected Cycles for Wafer Scale Integration*; Thinking Machines Corporation: Waltham, MA, USA, 1987; pp. 479–482.
21. Chen, J.L.; Shin, K.G. Addressing, routing, and broadcasting in hexagonal mesh multiprocessors. *IEEE Trans. Comput.* **1990**, *39*, 10–18. [[CrossRef](#)]
22. Robin, J.W.; John, J.W. *Graphs an Introductory Approach*; John and Wiley and Sons: Hoboken, NJ, USA, 1990.