Petersen-star networks modeled by optical transpose interconnection system

Jung-hyun Seo¹ and HyeongOk Lee²

Abstract
One method to create a high-performance computer is to use parallel processing to connect multiple computers. The structure of the parallel processing system is represented as an interconnection network. Traditionally, the communication links that connect the nodes in the interconnection network use electricity. With the advent of optical communication, however, optical transpose interconnection system networks have emerged, which combine the advantages of electronic communication and optical communication. Optical transpose interconnection system networks use electronic communication for relatively short distances and optical communication for long distances. Regardless of whether the interconnection network uses electronic communication or optical communication, network cost is an important factor among the various measures used for the evaluation of networks. In this article, we first propose a novel optical transpose interconnection system–Petersen-star network with a small network cost and analyze its basic topological properties. Optical transpose interconnection system–Petersen-star network is an undirected graph where the factor graph is Petersen-star network. OTIS–PSNₙ has the number of nodes 10²ⁿ, degree n + 3, and diameter 6n − 1. Second, we compare the network cost between optical transpose interconnection system–Petersen-star network and other optical transpose interconnection system networks. Finally, we propose a routing algorithm with a time complexity of 6n − 1 and a one-to-all broadcasting algorithm with a time complexity of 2n − 1.

Keywords
Optical transpose interconnection system, Petersen-star, interconnection network, network cost, graph

Date received: 24 January 2021; accepted: 19 June 2021
Handling Editor: Dr Yanjiao Chen

Introduction
Computing performance largely depends on processor speed and memory capacity. Parallel processing, which connects multiple processors, has emerged because improvements in the physical performance of processors and memory have relatively slowed down in recent years. For an interconnection network, the connection structure of the parallel processing system is represented on a topology. The interconnection network is represented as a graph mapping the processors as nodes and the communication links that connect the processors as edges.¹

Designing a good interconnection network is the same as creating a good parallel processing system. The interconnection networks that have been reported so far can be largely classified into mesh, torus, hypercube, and start graph classes. The hypercube class has been commercialized as iPSC/2, iPSC/860, and nCUBE/2; the mesh class as MasPar Intel Paragon, XP/
S, Touchstone DELTA System, and Mosaic C; and the three-dimensional mesh class as Cray T3D, MIT’s J-Machine, and Tera Computer. In the systems mentioned in the preceding paragraph, electronic signals are transmitted along the communication links. After the emergence of optical communication, an optical transpose interconnection system (OTIS), which combines the advantages of both electronic communication and optical communication, was proposed by Marsden et al. The OTIS network is implemented using free-space optoelectronic technology. In terms of transmission speed, power consumption, and error rate, the breakeven point for optical communication occurs when the length of the communication link is less than 1 cm. However, the installation cost is inexpensive for electronic communication.

In OTIS, \( n \) groups consisting of \( n \) nodes exist. Electronic links are used for connections between nodes in a group of relatively short distances, whereas optical links are used for connections between nodes existing in different groups. OTISs have a fast message transfer speed and low power consumption. The OTISs reported so far include OTIS-ops, OTIS-pops, OTIS-stack graph, OTIS-mesh, OTIS-torus, OTIS-hypercube, OTIS-star graph, OTIS-swapped network, OTIS-hyper hexa-cell, OTIS-biswapped network (BSN)-Mesh, BSN-Torus, BSN-MOT (mesh of tree), and BSN-hyper hexa-cell (HHC).

In a graph, the number of edges connected to a certain node is called the degree of the node, and the number of edges existing on the shortest path between two different nodes is called the distance. The degree of the graph is the maximum value among the degrees of all nodes. The diameter of the graph is the maximum value among the distances between arbitrary nodes. An important measure for evaluating interconnection networks is network cost, which can be calculated as the degree \( \times \) diameter. Because degree refers to the number of communication links, the hardware installation cost increases when the degree increases. The diameter represents the length of the message transfer path between two processors, and when the diameter increases, the message transfer time increases, thus affecting the software cost. Thus, network cost is an important measure for the evaluation of interconnection networks.

In this article, we propose an optical transpose interconnection system–Petersen-star network (OTIS–PSN) designed using a Petersen-star network (PSN), which has a small network cost. Section “OTIS, Petersen graph, and PSN” describes the basic concept of OTIS, the Petersen graph, and the PSN. Section “OTIS–PSN” defines the OTIS–PSN, analyzes its basic properties, and compares the network cost with other OTIS networks. Section “Communication algorithm” presents the routing and broadcasting algorithms for OTIS–PSN. Finally, section “Conclusion” concludes this article.

### OTIS, Petersen graph, and PSN

#### OTIS

A basic graph that constitutes OTIS is called a factor graph (hereinafter referred to as factor). An OTIS consists of \( n \) groups, and one group is the same as one factor. The processors inside a group are connected by electronic links called electronic edges (hereinafter referred to as e-edges). The processors in different groups are connected by optical links called optical edges (hereinafter referred to as o-edges).

OTIS networks are designed based on the OTIS rule (the \( p \) processor of the \( g \) group is connected to the \( g \) processor of the \( p \) group), which uses previously presented mesh, torus, hyper cube, and star graphs as factor graphs. Furthermore, the biswapped family is designed to double the OTIS network and modify an edge. Because OTISs outperform previously announced interconnection networks theoretically, they have garnered significant research interest in recent times.

The address of node \( p \) in group \( g \) is denoted by \( g, p \). According to the OTIS rule, \( p, g \), and \( g, p \) are connected. Node \( p \) uses the node address of the factor, and thus \( g \) also uses the node address of the factor because \( g \) is exchanged with \( p \).

In OTIS, the number of groups is exactly the same as the number of nodes of the factor. Suppose that the

![Figure 1. OTIS-Q2](image-url)
The number of nodes in the factor is \( N \). Then, the number of nodes in the OTIS is \( N^2 \). The number of nodes of OTIS-Q2 in Figure 1 is 16 (=4\(^2\)).

The outline of OTIS routing is as follows. The starting node is \( S = (g_1, p_1) \), and the terminal node is \( T = (g_2, p_2) \). The path using the e-edge is represented as \( \rightarrow \), and the path using o-edge as \( \Rightarrow \). If \( g_1 = g_2 \), then the routing between the two nodes follows the routing of the factor. If \( g_1 \neq g_2 \), then the routing path between the two nodes is

\[
\langle g_1, p_1 \rangle \rightarrow \langle g_1, g_2 \rangle \Rightarrow \langle g_2, g_1 \rangle \rightarrow \langle g_2, p_2 \rangle
\]

**Petersen graph**

The Petersen graph consists of 10 nodes and 15 edges. It has the best network cost among graphs with 10 nodes, and has a degree of 3 with a diameter of 2.

Figure 2 shows the Petersen graph. The number on the left in the node address is smaller than the number on the right.

In a Petersen graph, \( P = (V_p, E_p) \); Node \( V_p = xy; \) Edge \( E_p = (xy, x'y'); x, y \in \{1,2,3,4,5\}; \) and \( x'y' \in \{\{1,2,3,4,5\} - \{x, y\}\}. \) It is assumed that in the Petersen graph, node \( u = u_1u_2 \) is the start node and node \( v = v_1v_2 \) is the destination node. Node \( w \) is composed of \( \{u_1, u_2\} \cup \{v_1, v_2\} \). The routing algorithm from \( u \) to \( v \) is shown in Algorithm 1.

For the all-port model, the one-to-all broadcasting in the Petersen graph is edge-disjoint and broadcasting time is 2. Initially, node 12 has a message. As presented in Figure 2(b), the broadcasting algorithm given in Algorithm 2 is applied.

Because the Petersen graph is a node (edge)-symmetric graph, Algorithm 2 is valid even though any arbitrary node can have a message initially.

**PSN**

The PSN uses the Petersen graph as the basic module. The number of internal edges that are connected to one node is three and the number of external edges is \( n - 1 \). PSN\(_n\) is a graph with \( 10^n \) as the number of nodes, \( n + 2 \) as the degree, and \( 3n - 1 \) as the diameter.

PSN\(_n\) = (\( V_p, E_p \)). The node of PSN\(_n\) is defined below

\[
V_p = \{p_1p_2p_3 \cdots p_{i-1}p_ip_i+1 \cdots p_{2n-1}p_{2n}| (p_ip_{i+1}, i \text{ is odd}) \text{ is the node address of the Petersen graph}\}
\]

The edges of PSN\(_n\) can be divided into internal and external edges. The internal edges connect nodes belonging to the same Petersen graph. The external edges connect nodes belonging to different Petersen graph. An external edge connects two nodes that have exchanged the symbols of \( p_ip_{i+1} \) of the node.
In Figure 3, the numbers in black are the addresses with each other. The external edge of PSN is defined as:

\[ \text{Edge } E_i = (p_1p_2p_3 \ldots p_{i-1}p_ip_i+1 \ldots p_{2n-1}) \]

where \( 2 < i < 2n \) and \( i \) is odd.

In Figure 3, the numbers in black are \( p_1p_2 \) and the numbers in red are \( p_3p_4 \). For example, nodes 1245 and 1234 are connected to nodes 4512 and 3412, respectively, through external edges. In Figure 3, the external edges are indicated by blue solid lines, and the internal edges are indicated by black solid lines.

It is assumed that in PSN, node \( u = u_1u_2u_3 \ldots u_i \ldots u_t \) is the start node and node \( v = v_1v_2v_3 \ldots v_j \ldots v_s \) is the destination node. The routing algorithm from \( u \) to \( v \) is as given in Algorithm 3.23

The time complexity of Algorithm 3 is as follows. In terms of the worst case in each line, the time complexity is \( 2 \) in lines 2 and 5, and \( 1 \) in line 3. Therefore, the time complexity is \( (n-1) \times (2+1) + 2 = 3n - 1 \). This value is the diameter of PSN. For example, if \( u = 12341345 \) and \( v = 35241525 \) in PSN, the routing path is as follows:

\[
\begin{align*}
(\text{u}) & \rightarrow 12341345 \rightarrow 25341325 \rightarrow 45341325 \rightarrow 15341325 \\
& \rightarrow 15341325 \rightarrow 13341525 \rightarrow 24341525 \\
& \rightarrow 34241525 \rightarrow (\text{v}) 35241525
\end{align*}
\]

### OTIS–PSN

**Definition**

OTIS–PSN is divided into \( 10^n \) groups, where each group has \( 10^n \) nodes. OTIS–PSN is defined as follows:

\[ V = \{ <x,y> \mid x, y \in Vp \mid Vp \text{ is a node of PSN} \} \]

The edges in OTIS–PSN are divided into o-edges and e-edges. The e-edges connect the edges within PSN. The o-edges of the OTIS–PSN are defined as follows:

\[ \text{o-edge } = ( <x,y> , <y,x> \mid x \neq y) \]

Figure 4 shows OTIS–PSN with the factor PSN. Each group is represented by a blue circle, and the group address is expressed as an orange four-digit number on the circle. OTIS–PSN consists of \( 10^2 \) groups. One group is shown in detail on the left. Only selected groups (out of 99) are shown on the right to limit the complexity of the diagram. Orange lines represent the o-edges, and black lines represent the e-edges. Again, only certain edges are shown to limit complexity.

For example, let us look at node \( <1212, 1234> \) in Figure 4 (left). \( <1212, 1234> \) is connected to \( <1234, 1212> \) by an e-edge and connected to \( <1212, 1212>, <1212, 1225>, <1212, 1215>, \) and \( <1212, 3412> \) by e-edges. Let us examine the nodes connected to node \( <1212, 1234> \) by e-edges. Node \( <1212, 3412> \) is connected by an external edge, and the remaining three nodes are connected by internal edges. In Figure 4 (left), which is a group, 99 nodes excluding node \( <1212, 1212> \) are all connected to nodes belonging to other groups. Similarly, nodes \( <x, y> \) belonging to the remaining 99 groups are connected to different groups, excluding the case where \( x = y \).

### Topological properties of OTIS–PSN

In this section, the basic topological properties of OTIS–PSN are examined. The number of nodes in OTIS–PSN is shown by Theorem 1.
Theorem 1. The total number of nodes in OTIS–PSN, is $102n$.

Proof. The factor of OTIS–PSN, is PSN, the number of nodes in PSN, is $10n$. OTIS–PSN consists of $10n$ groups, and each single group is a PSN. Therefore, the total number of nodes is $102n = (10n)^3$. 

The degree of an arbitrary node is the number of edges incident to that node, and the degree of the graph is the maximum value among the degrees of all nodes. The degree of OTIS–PSN is shown by Theorem 2.

Theorem 2. The degree of OTIS–PSN, is $n + 3$.

Proof. The degree of PSN, is $n + 2$. The number of e-edges connected to an arbitrary node $<x, y>$ of OTIS–PSN, is $n + 2$. The number of o-edges connected to an arbitrary node $<x, y>$ is 0 when $x = y$ and 1 when $x \neq y$. Therefore, the degree is $(n + 2) + 1 = n + 3$.

In a graph, a path is a node sequence that connects one node to another node, and distance is the number of edges on that path. Between two arbitrary nodes, the minimum distance is $d$, and the largest $d$ when accounting for all nodes is referred to as the diameter of graph. The diameter of OTIS–PSN is shown by Theorem 3.

Theorem 3. The diameter of OTIS–PSN, is $6n - 1$.

Proof. Suppose that two arbitrary nodes are $u = <x_1, y_1>$ and $v = <x_2, y_2>$. Creating a path between $u$ and $v$ is the same as changing the address of $u$ to that of $v$. The address change can be performed by only the e-edge and o-edge. There are two main methods for changing the address of $u$ to the address of $v$. The first method changes $x_1$ to $x_2$ and $y_1$ to $y_2$. The second method changes $x_1$ to $y_2$ and $y_1$ to $x_2$. When a change of address is made, the worst-case time complexity is $3n - 1$. This is because the diameter of PSN, is $3n - 1$. The second method is used to set the shortest path between the two nodes. This path is as follows: $<x_1, y_1> \rightarrow <x_1, x_2> \rightarrow <x_2, x_1> \rightarrow <x_2, y_2>$. The length of this path is $2(3n - 1) + 1$. Therefore, the diameter is $6n - 1$.

Network cost is an important measure for the evaluation of a network. Table 1 shows the network cost of OTIS–PSN versus various OTIS networks. For ease of understanding, we compare network costs per node, as $N$ is the node quantity in Table 1. To calculate the network cost of other networks, the diameter and degree of OTIS-mesh, OTIS-torus, OTIS-swapped network, OTIS-biswapped network, OTIS-hypercube, and OTIS-hyper hexa-cell—were investigated. The OTIS-biswapped families—BSN-Mesh, BSN-Torus, BSN-MOT, BSN-hhc—are omitted from Table 1, and only OTIS-biswapped network is shown because biswapped families have similar measures. The network cost is shown by the notation $O$.

As shown in Table 1, because $O((\log_{10} \sqrt{N})^2) < O((\log_2 \sqrt{N})^2) < O(\sqrt{N}) < O(\sqrt{N})$, OTIS–PSN is superior to other OTIS networks in terms of network cost. Table 2 shows the network costs of Table 1 by the number of nodes.

Theorem 4. OTIS–PSN, is a connected graph.

Proof. First, it should be determined whether PSN, which forms a group in OTIS–PSN, is a connected
These 10 nodes are connected to 10 nodes, connected to node Y is connected nodes is denoted by $y_i$. Nodes with the same $X$ but different $Y$ form a group. Suppose that $Y = y_1y_2y_3...y_i - 1 y_i y_i + 1 y_2 y_2 - 1 y_2 y_2$. According to the definition of PSN, 10 nodes with different $y_1 y_2$ are all connected in a Petersen graph, as shown in Figure 2. A set of all connected nodes is denoted by $CC$. The set of 10 nodes connected to node Y is $CCy_3...y_i - 1 y_i y_i + 1 ... y_2 y_2 - 1 y_2 y_2$. These 10 nodes are connected to 10 nodes, $y_3 y_4 y_5 y_6 y_7 y_8 y_9 y_10$ by $E_3$. These 10 nodes are connected to 100 nodes $CC...CC...CC$ in the same group are connected. Therefore, the factor $PSN_n$ of OTIS is a connected graph. A node $<X, Y>$ is connected to a node $<Y, X>$ by an o-edge. In $<Y, X>$, $Y$ indicates all groups. Therefore, all the nodes of the OTIS–PSN are connected.

**Communication algorithm**

**Routing algorithm**

The routing path between two nodes is the message delivery path. A path is a node sequence connected by an edge. In OTIS–PSN, the starting node is $u = <ug, up>$, and the destination node is $v = <vg, vp>$. Routing is the process of matching the address of node $u$ with that of node $v$. The node address can be changed by the e-edge and o-edge. $up$ is changed to $vg$, and $ug$ to $vp$. An arbitrary intermediate node on the routing path at a certain point of time is called $w = <wg, wp>$. The routing algorithm from $u$ to $v$ is as given in Algorithm 4.

Because the worst-case time complexity of Algorithm 3 is $3n - 1$, the worst-case time complexity of Algorithm 4 is $2(3n - 1) + 1 = 6n - 1$. Although Algorithm 4 is not optimal, the length of the path is less than or equal to the diameter in Theorem 3. For example, if the starting node $u$ is $<134524, 152412>$ and the destination node $v$ is $<352313, 123425>$ in OTIS–PSN, the routing path is as follows:

| $u$ | $<134524, 152412>$ | $<134524, 132412>$ |
|-----|-----------------|-----------------|
| $\rightarrow$ | $<134524, 124313>$ | $<134524, 234213>$ |
| $\rightarrow$ | $<134524, 234213>$ | $<134524, 242313>$ |
| $\rightarrow$ | $<134524, 352313>$ | $= >$ |
| $\rightarrow$ | $<352313, 134524>$ | $<352313, 254524>$ |
| $\rightarrow$ | $<352313, 244525>$ | $<352313, 344525>$ |
| $\rightarrow$ | $<352313, 344525>$ | $<352313, 453425>$ |
| $\rightarrow$ | $<352313, 123425>$ | $v$ |

---

**Table 1.** Comparison of network cost with other OTIS networks.

| Network                     | Degree | Diameter | Network cost                          |
|-----------------------------|--------|----------|---------------------------------------|
| OTIS-mesh $M (\sqrt{N}, \sqrt{N})$ | 5      | $4\sqrt{N} - 3$ | $O(\sqrt{N})$ |
| OTIS-torus $T (\sqrt{N}, \sqrt{N})$ | 5      | $2\sqrt{N} - 3$ | $O(\sqrt{N})$ |
| OTIS-swapped network $Sk_n$ | $\sqrt{N}$ | 3        | $O(\sqrt{N})$ |
| OTIS-biswapped network $BSk_n$ | $0.5\sqrt{N}$ | 1.5      | $O(\sqrt{N})$ |
| OTIS-hypercube $Q_n$        | $\log_2 \sqrt{N} + 1$ | $2\log_2 \sqrt{N} - 2$ | $O((\log_2 \sqrt{N})^2)$ |
| OTIS-hhc, $n$               | $\log_2 0.17 \sqrt{N} + 4$ | $2\log_2 0.17 \sqrt{N} + 5$ | $O((\log_2 \sqrt{N})^2)$ |
| OTIS-Petersen-star OTIS–PSN | $\log_2 0.17 \sqrt{N} + 3$ | $6\log_2 0.17 \sqrt{N} - 1$ | $O((\log_2 \sqrt{N})^2)$ |

**Table 2.** Network cost by number of nodes.

| Network | Node number | $10^2$ | $10^3$ | $10^4$ | $10^5$ | $10^6$ | $10^7$ | $10^8$ |
|---------|-------------|--------|--------|--------|--------|--------|--------|--------|
| OTIS-mesh, torus | 3.16 | 5.62 | 10.00 | 17.78 | 31.62 | 56.23 | 100.00 |
| OTIS-swapped, biswapped | 10.00 | 31.62 | 100.00 | 316.23 | 1000 | 3162.28 | 10,000.00 |
| OTIS-hypercube, hhc | 11.04 | 24.83 | 44.14 | 68.97 | 99.32 | 135.18 | 176.56 |
| OTIS–Petersen-Star OTIS–PSN | 1.00 | 2.25 | 4.00 | 6.25 | 9.00 | 12.25 | 16.00 |

OTIS: optical transpose interconnection system.
Broadcasting algorithm

Routing transfers a message from the node that has the message to another node. In contrast, broadcasting sends a message to every node in the network. Broadcasting is performed by successive calling to transmit a message and has several limitations: first, only two nodes participate in a call; second, a call is completed in a unit time; third, each node may take part in only one call in a unit time; and fourth, a call appears only between two neighboring nodes. In one-to-all broadcasting, the originating node sends the message to all nodes, whereas in all-to-all broadcasting, each node has a different message, which it sends to all nodes. Here, we use one-to-all broadcasting. The one-port model allows a node to send a message to only one other node in a unit of time, whereas the multi-port model allows a node to send a message to multiple nodes. Here, we deal with the all-port model, which allows a message to be sent to all nodes in the same unit time.

The following is a brief overview of the broadcasting algorithm:

Stage 1. The message is sent to all nodes of the factor graph \( f \) to which belongs the node in which the message originated.
Stage 2. The message is transmitted to all the nodes that are connected to the nodes of \( f \) by \( o \)-edges. In each group, a single node receives the message.
Stage 3. In each group, Stage 1 is performed to simultaneously deliver the message to all nodes in the same group.

Next, to better understand the broadcasting algorithm, the broadcasting in OTIS–PSN\( \text{N}_2 \) is examined. The node that initially had the message is called \( u = \langle g, p \rangle \). Here, \( g = g_1 g_2 g_3 g_4, p = p_1 p_2 p_3 p_4 \). \( g_1 = 12, g_2 = 34, g_3 = 45, g_4 = 23 \). Here, \( xx = \{12, 13, 14, 15, 23, 24, 25, 34, 35, 45\} \). The values that \( g_1 g_2, g_3 g_4, p_1 p_2, \) and \( p_3 p_4 \) can take. Thus, \( |xx| = 10 \). In Stage 1, a node \( u = \langle 1234, 2345 \rangle \) transmits a message to a node set \( \langle 1234, xx45 \rangle \) according to Algorithm 2. Here, the number of nodes with the message is 10. Node set \( \langle 1234, xx45 \rangle \) then sends the message to node set \( \langle 1234, 45xx \rangle \), connected by \( E_3 \). Here, the number of nodes with the message is 20. The node set \( \langle 1234, 45xx \rangle \) sends the message to another node set \( \langle 1234, xxxx \rangle \) according to Algorithm 2, and Stage 1 is terminated. Here, the number of nodes having the message is 100, and every node of factor \( \langle 1234, xxxx \rangle \), to which \( u \) with the initial message belongs, has the message.

Stage 2 is simple. A node set \( \langle 1234, xxxx \rangle \) sends the message to node sets \( \langle xxxx, 1234 \rangle \) that are connected to each node by \( o \)-edges, and Stage 2 is terminated. Here, the number of nodes having the message is 200, and at least one node in every group has the message. In Stage 3, in each group, the node \( \langle xxxx, 1234 \rangle \), which has the message from stage 1, sends the message simultaneously to all nodes \( \langle xxxx, xxxx \rangle \) in

Algorithm 5.

Broadcasting (\( u \) )

Stage 1

1: \(<g_1g_2g_3g_4 \in \{12,13,14,15,23,24,25,34,35,45\}, p_1p_2p_3p_4 \in \{1\ldots p_2n-1\ldots p_2n\} \rangle>\)
2: \(<g_1g_2g_3g_4 \in \{12,13,14,15,23,24,25,34,35,45\}, p_1p_2p_3p_4 \in \{1\ldots p_2n-1\ldots p_2n\} \rangle>\)
3: \(<g_1g_2g_3g_4 \in \{12,13,14,15,23,24,25,34,35,45\}, p_1p_2p_3p_4 \in \{1\ldots p_2n-1\ldots p_2n\} \rangle>\)
4: Lines 2 and 3 are repeated \((n/2) - 2\) times, which can be expressed as follows:
5: \(<g_1g_2g_3g_4 \in \{12,13,14,15,23,24,25,34,35,45\}, p_1p_2p_3p_4 \in \{1\ldots p_2n-1\ldots p_2n\} \rangle>\)

Stage 2

1: \(<g_1g_2g_3 \in \{12,13,14,15,23,24,25,34,35,45\}, i \in \{1\ldots g_2n-1\ldots g_2n\} \rangle>\)
2: \(<xxxxx\ldots xxxx \in \{12,13,14,15,23,24,25,34,35,45\}, g_1 \in \{1\ldots g_2n-1\ldots g_2n\} \rangle>\)

Stage 3

1: Stage 1 is performed at every node simultaneously, which can be expressed as follows:
2: \(<xxxxx\ldots xxxx \in \{12,13,14,15,23,24,25,34,35,45\}, i \in \{1\ldots g_2n-1\ldots g_2n\} \rangle>\)
3: \(<xxxxx\ldots xxxx \in \{12,13,14,15,23,24,25,34,35,45\}, i \in \{1\ldots g_2n-1\ldots g_2n\} \rangle>\)
4: \(<xxxxx\ldots xxxx \in \{12,13,14,15,23,24,25,34,35,45\}, i \in \{1\ldots g_2n-1\ldots g_2n\} \rangle>\)
the factor it belongs to. This can be generalized to create a broadcasting algorithm, as shown in Algorithm 5. The node that has the message first is called $u = \langle g, p \rangle$, $p = p_1p_2p_3 \ldots p_i - 1p_1p_2 \ldots p_{2n} - 1$, and $\rightarrow$ refers to message transmission.

**Theorem 5.** In OTIS–PSN$_n$, the time complexity for the one-to-all broadcasting algorithm of the all-port model is $3n - 7$.

**Proof.** The time complexity of broadcasting Algorithm 2 is calculated. In Steps 1.1 and 1.3, the message is sent to 10 nodes of the Petersen graph to which it belongs, in only two times, according to broadcasting Algorithm 1. Step 1.2 is performed once. In Step 1.4, Steps 1.2 and 1.3 are repeated $(n/2) - 2$ times. The time complexity of Stage 1 is $2 + (n/2 - 2) \times (2 + 1) = 1.5n - 4$. The time complexity of Stage 2 is 1. Because Stage 3 is the same as Stage 1, the total time complexity is $2(1.5n - 4) + 1 = 3n - 7$. □

**Conclusion**

In this article, we proposed an OTIS–PSN that has a small network cost and analyzed its basic topological properties. We then compared the network cost of previously reported OTIS networks with the proposed OTIS network (OTIS–PSN). OTIS–PSN$_n$ has $10^2n$ number of nodes, $n + 3$ degree, and $6n - 1$ diameter. If the number of nodes is $N$, the network cost of OTIS–PSN is $O((\log_{10}\sqrt{N})^2)$. This is superior to the network cost $O(\log_2\sqrt{N}^2)$ of OTIS-hypercube, $O(\sqrt{N})$ of OTIS-torus, and $O(\sqrt{N})$ of the OTIS-swapped network. We also proposed a routing algorithm with a time complexity of $6n - 1$, and a one-to-all broadcasting algorithm with a time complexity of $2n - 1$. In OTIS networks, optical communication and electronic communication are used together for communication links. Further research should be conducted by applying the proposed algorithms to connected networks separately for electronic communication and optical communication, or particularly for optical communication.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (no. 2020R1A2C1012363).

**ORCID iD**

Jung-hyun Seo https://orcid.org/0000-0003-2117-5750

**References**

1. Seo JH and Lee HO. Design and analysis of the rotational binary graph as an alternative to hypercube and torus. J Supercomput 2020; 76(9): 7161–7176.
2. Seo JH, Kim JS, Chang HJ, et al. The hierarchical Petersen network: a new interconnection network with fixed degree. J Supercomput 2018; 74(4): 1636–1654.
3. Marsden GC, Marchand PJ, Harvey P, et al. Optical transpose interconnection system architectures. Optic Lett 1993; 18(13): 1083–1085.
4. Zane F, Marchand PJ, Patrui R, et al. Scalable network architectures using the optical transpose interconnection system. In: Proceedings of the third international conference on massively parallel processing using optical interconnections, Maui, HI, 27–29 October 1996, pp. 114–121. New York: IEEE.
5. Coudert D, Ferreira A and Muñoz X. Topologies for optical interconnection networks based on the optical transpose interconnection system. Applied Optics 2000; 39(17): 2965–2974.
6. Wang CF and Sahni S. Basic operations on the OTIS-mesh optoelectronic computer. IEEE Trans Parallel Distrib Syst 1998; 9(12): 1226–1236.
7. Lucas KT and Jana PK. Parallel algorithms for finding polynomial roots on OTIS-torus. J Supercomput 2010; 54(2): 139–153.
8. Mahafzah BA and Jaradat BA. The load balancing problem in OTIS-hypercube interconnection networks. J Supercomput 2008; 46(3): 276–297.
9. Awwad AM. OTIS-star an attractive alternative network. In: Proceedings of the 4th WSEAS International Conference on Software Engineering, Parallel & Distributed Systems. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, WI, 13 February 2005. New York: ACM.
10. Li HX, Ahmad S and Ahmad I. Topology-Based Analysis of OTIS (Swapped) Networks OKn and OPn. J Chem 2019; 2019: 1–11.
11. Mahafzah BA, Sleit A, Hamad NA, et al. The OTIS hyper hexa-cell optoelectronic architecture. Computing 2010; 94(5): 411–432.
12. Wei W, Li Q and Tao M. BSN-mesh and its basic parallel algorithms. Int J Grid Util Comput 2015; 6(3–4): 213–220.
13. Gupta A and Sarkar BK. Biswapped–Torus network: a new efficient node-symmetrical optoelectronic network. Natl Acad Sci Lett 2021; 44(1): 27–31.
14. Gupta A and Sarkar BK. BSN MOT: a fast and cost-efficient optoelectronic architecture. IETE Technical Review. Epub ahead of print 27 September 2020. DOI: 10.1080/02564602.2020.1823251.
15. Gupta A and Sarkar BK. The recursive and symmetrical optoelectronic network architecture: biswapped network hyper hexa-cell. Arab J Sci Eng 2018; 43(12): 7635–7653.
16. Parhami B and Yeh CH. Why network diameter is still important. In: Proceedings of international conference on
communication in computing, Las Vegas, NV, 26–29 June 2000, pp. 271–274. New York: ACM.

17. Ahmad A, Hasni R, Elahi K, et al. Polynomials of degree-based indices for swapped networks modeled by optical transpose interconnection system. IEEE Access 2020; 8: 214293–214299.

18. Zahra N, Ibrahim M and Siddiqui MK. On topological indices for swapped networks modeled by optical transpose interconnection system. IEEE Access 2020; 8: 200091–200099.

19. Cancan M, Ahmad I and Ahmad S. Molecular descriptors of certain OTIS interconnection networks. Proyecciones 2020; 39(4): 769–786.

20. Al-Adwan A, Zaghoul R, Mahafzah BA, et al. Parallel quicksort algorithm on OTIS hyper hexa-cell optoelectronic architecture. J Parallel Distrib Comput 2010; 141: 61–73.

21. Gupta A, Singh H and Bhati A. Efficient parallel algorithm for mapping LaGrange’s interpolation on OTIS and BSN hyper hexa-cell. In: 2020 international conference on emerging smart computing and informatics (ESCI), Pune, India, 12–14 March 2020. New York: IEEE.

22. Memmi G and Raillard Y. Some new results about the (d, k) graph problem. IEEE Trans Comput 1982; c-31(8): 784–791.

23. Seo JH and Lee HO. Design and analysis of the Petersen star interconnection network. Far East J Electron Commun 2017; 17(4): 719–732.

24. Seo JH and Lee HO. Link-disjoint broadcasting algorithm in wormhole-routed 3D Petersen-Torus networks. Int J Distrib Sens Netw 2013; 9(12): 1–7.