IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 15, 2021, accepted June 25, 2021, date of publication July 6, 2021, date of current version July 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095146

The Static Performance Effect of Hybrid-Hierarchical Interconnection by Shifted Completely Connected Network

MOHAMMED N. M. ALI[1], M. M. HAFIZUR RAHMAN[2], ADAMU ABUBAKAR IBRAHIM[1], (Member, IEEE), YASUSHI INOGUCHI[3], (Member, IEEE), AND EKLAS HOSSAIN[4], (Senior Member, IEEE)

[1]Department of Computer Science, KICT, International Islamic University Malaysia (IIUM), Kuala Lumpur 53100, Malaysia
[2]College of Computer Science and Information Technology (CCSIT), King Faisal University, Al Ahsa 31982, Saudi Arabia
[3]Research Center for Advanced Computing Infrastructure, JAIST, Nomi-Shi, Ishikawa 923-1211, Japan
[4]Department of Electrical Engineering and Renewable, Oregon Renewable Energy Center (OREC), Oregon Tech, Klamath Falls, OR 97601, USA

Corresponding author: Mohammed N. M. Ali (moh.ali.exe@gmail.com)

ABSTRACT Massively parallel computer (MPC) systems execute many operations based on internal networks called interconnection networks. The performance of these networks is affected by their topologies.There are many topologies of interconnection networks for MPC systems, unfortunately, they faced many drawbacks. Expanding the size of the network degrades the performance of these topologies. That is why this current paper presents a hybrid-hierarchical interconnection network (HIN) topology by Shifted Completely Connected Network (SCCN) to circumvent the drawbacks of the existing topologies. An experimental evaluation involving the design and development of a hierarchical network was carried out. A two-dimensional higher level networks has been produced and its static network performance parameters evaluated through simulators. The finding of the simulations has shown some good performances compared to many previous designed networks. SCCN is better than all conventional networks in terms of diameter, cost and average distance.

INDEX TERMS Shifted completely connected network (SCCN), Network-on-Chip, interconnection networks, hierarchical interconnection networks (HINs), static network performance parameters, massively parallel computer (MPC) systems.

I. INTRODUCTION

Day by day, the need for supercomputer systems is increasing; these systems are used to process an immense volume of data rapidly. Besides, these systems solve complicated problems in parallel by dividing the problem into parts and distributing these parts between thousands of CPUs and then combining the results to provide an optimal and fast solution. The importance of these systems is elevated because they provide the power to collect, organize, and analyze big amounts of data. Also, they have many benefits which will benefit the modern life of humankind by developing new energy sources, improving the medical services, avoiding disasters, forecasting the weather, and many other beneficial uses [1]. The high demand for signaling technology makes it difficult for the systems with a single CPU to satisfy this need. This participates in making the current computing problems difficult to be handled by sequential computers [2]. This motivated the research community to find new solutions to fulfil the new technology requirements. Therefore, multiprocessor systems have emerged to replace the sequential computers. Massively parallel computer (MPC) systems are considered as the highest-level computers with the ability to model many problems in various areas [3]. Improving the performance and reducing the cost of MPC systems is the priority of the research work. Therefore, many designs of MPC systems have been presented looking for an ideal one.

The underlying interconnection network is the backbone of MPC systems; these networks are responsible for interconnecting the processing elements (PEs) inside the system, and

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

have a vital role in either improving or degrading the system performance [1], [4]. Besides, they play the main role in determining the cost. Therefore, many research-works focus on improving the performance of these networks by presenting different topologies to arrange and manage the connection between the PEs inside the system [5]. The performance of these topologies have been evaluated statically and dynamically to examine the effectiveness of these topologies. The main goal of the evaluation is finding an optimal topology to design MPC systems. Thus, many topologies were proposed, but it is still infeasible to be implemented [6]. Expanding the interconnection network size by increasing the number of computing nodes is the most important factor to strengthen the computing power of the system which will enable it to overcome the increasing demand of the computational power. The early structures of the interconnection network topologies showed poor performance with the increase of the network size [7]. Therefore, to solve this problem, hierarchical interconnection networks (HINs) were proposed as an alternative solution to replace the conventional networks in building MPC systems [8]. These networks proved their capability in expanding the number of computing nodes inside the system to millions of nodes. Furthermore, HINs are affordable and fault tolerance networks for MPC systems and proficient in reducing cost and power consumption [9].

Network-on-Chip (NoC) emerged as a single silicon chip to be used in employing the communication structures of large-scale to very-large-scale integration (VLSI) systems. The rapid improvements of VLSI design have created a suitable environment to place a complete system in a single chip. The benefits of using NoC in designing the large-scale systems are reducing the system wires complexity, controlling the power, providing a reliable system; besides, it offers high flexibility and regular network structure [10]. The performance of NoC is affected by the design of the network topology. Also, the correct choice of routing protocol is important in decreasing the network latency and congestion. The architecture and the performance of MPC system is predicted to have a promising future depending on NoC [11]. TrueNorth is unveiled by IBM; it is a chip architecture inspired by the human brain's function and efficiency. This chip is created from an interconnected network of neurosynaptic cores. TrueNorth is completely programmable in terms of anatomy and physiology of the chip, which means a spacious limit of dynamics, behaviors and structures are allowed by synaptic crossbar, neuron parameters and inter-core neuron-axon connectivity [12]. Using this technology in building neurosynaptic supercomputers by utilizing numerous TrueNorth chips will make a revolution in this field [13].

In this paper, a new HIN is proposed to build an MPC system by connecting various levels in a hierarchical design. This system is built based on a topology called shifted completely connected network (SCCN) and is composed of different levels including the Chip-Level, the Board-Level, the Cabinet- Level, and the System-Level. This network is proposed to solve the problem of expanding the size of MPC systems which emerged by using the conventional topologies in building MPC systems such as 2D-mesh, 2D-torus, hypercube, and so on. Expanding the system size based on these networks will reduce the performance of these systems. Thus, in this paper, the size of the proposed network is expanded by proposing different levels of SCCN and the static performance of each level is compared with other networks with almost the same number of nodes.

The rest of the paper is organized as follows. Section II describes the architecture of the proposed network. Section III explains the method of addressing the nodes of the network. Section IV elaborately discusses the shortest path routing algorithm proposed for the SCCN. Next, Section V discusses the static network performance parameters of the SCCN in comparison with the other conventional networks. Thereafter, Section VI provides some generalizations based on the proposed network. Then, Section VII contains the limitations of this work and highlights some future work directions. Finally, Section VIII concludes the paper.

II. ARCHITECTURE OF THE PROPOSED SYSTEM

This section discussed the architecture of a parallel computer system built based on the SCCN topology. The architectures of the Chip-Level, Board-Level, Cabinet-Level, and System-Level networks are elaborated here. Different structures networks with extra numbers of nodes are presented; each higher-level is the connection of multiple lower-levels in a recursive design.

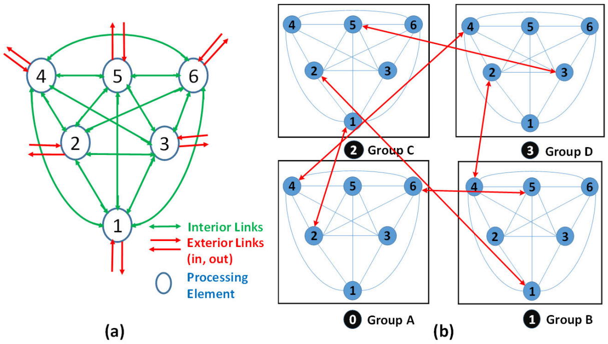

FIGURE 1. **The Architecture of (a) Chip-Level network built based on a network of six nodes connected completely and is known as Shifted Completely Connected Network (SCCN). It is considered as a basic module (BM) of a parallel computer system; (b) The Chip-Level network composed of multiple BMs connected in a (2 × 2) network and it divided into two sets of nodes in x-direction and two sets of nodes in y-direction.**

A. CHIP-LEVEL NETWORK

The Chip-Level network is based on a network of six nodes connected completely and called Shifted Completely Connected Network (SCCN) as shown in Figure 1(a). This topology considers as a basic module (BM) of a parallel computer system. Multiple BMs are connected in hierarchical design based on a shifted mechanism to create the Chip-Level network. Thus, to build a system with a greater number of nodes; we will expand the size of SCCN to reach ten or a hundred thousand nodes. The connectivity and the design of the BM-Level of SCCN is shown in Figure 1(a), which is composed of six nodes connected completely and the connection between the nodes take place in two directions; the details are discussed in previous work [14]. Also, it is evaluated and compared to different networks. Each node in the BM-level has additional links to act as gate nodes for external connections. The Chip-Level network is composed of multiple BMs connected in a (2 × 2) network and it divided into two sets of nodes in x-direction and two sets of nodes in y-direction as shown in Figure 1(b), the connection between the BMs established by shifting the binary digits of the index of each node steps to the left or to the right.

B. BOARD-LEVEL NETWORK

The board-Level is composed of multiple BMs connected hierarchically; each BM is a ($2^m \times 2^m$) network and refers to the NoC of SCCN. 1(b) illustrates the architecture of the BM of the Board-Level network; there are 4 groups connected through global bidirectional links. Each group has numbers of free ports to be used in creating advance level networks. Multiple NoC networks are connected recursively in a ($2^m \times 2^m$) 2D-torus network, where m is a positive integer number. 2D-torus is used to expand this system because it has features summarized in the high speed, less cost, low latency, better fairness, and low power consumption.

Considering ($m = 2$) will provide a Board-Level network with $2^2 \times 2^2 = 4 \times 4 = 16$ BMs laying in two dimensional rectangular and divided into 2^m rows and 2^m columns. Each node from this level is connected to its nearest neighbors, and the connection between them take place in 4 directions; these directions are +X, −X, +Y, −Y. However, the BMs in the edges are connected through wrap-around links. The benefit of using the wrap-around links in this network is to decrease the distances between the edges, which will provide a fair connection between the BMs of this network.

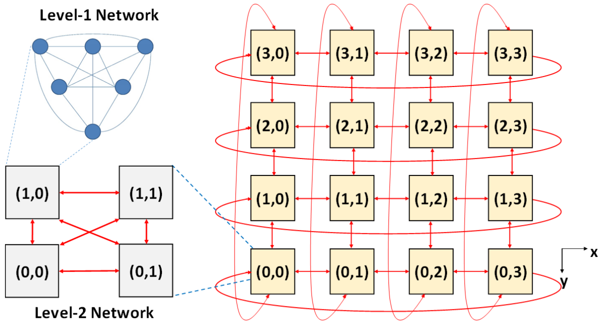

FIGURE 2. The architecture of a (4 × 4) Board-Level Network. The BMs in the edges connected through wraparound links for which the distances between the edges decrease, providing a fair connection between the BMs of this network.

The architecture of a (4 × 4) Board-Level illustrated in Figure 2. The total number of nodes in the higher levels of SCCN can be obtained from Equation 1.

$$N_L = N_{L-1} \times 2^m, \quad (1)$$

where N_L is the number of processing elements (PEs) in Level (L), N_{L-1} is the number of PEs Level (L-1), and m is a positive integer. From Equation 1, the total number of nodes in the Board-Level is $N_{L2} = N_{L1} \times 2^m$, where N_{L1} is the number of PEs in the Chip-Level which equal to 24 nodes. The value of m can be any number, however, $m = 2$ will be considered to create a (4 × 4) network. By replacing these values in Equation 1, the number of PEs in the Board-Level is equal to 384 nodes.

C. CABINET-LEVEL NETWORK

Expanding the size of a supercomputer system is important in creating a device with hundreds of thousands or millions of processing elements (PEs); these PEs can process massive data in parallel and faster. Therefore, we will focus in this section on proposing larger network than the previous level called the Cabinet-Level. The BM of the Cabinet-Level is a (4 × 4) network and refers to the Board-Level which we have discussed earlier. The Cabinet-Level network is composed of several BMs connected in a (4 × 4) 2D-torus network in hierarchical design. The nodes in this level connected to the neighbor nodes in four directions and the nodes in the edges using wrap-around links for connection.

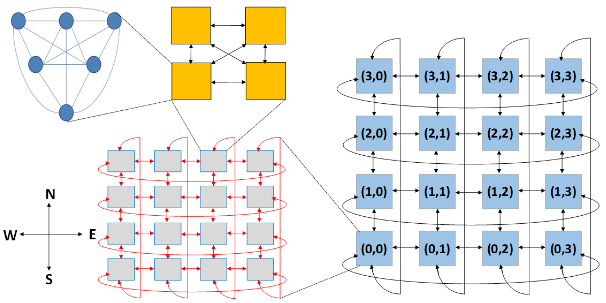

FIGURE 3. The structural design and connectivity of a (4 × 4) Cabinet-Level network. It is composed of several BMs connected in a (4 × 4) 2D-torus network in a hierarchical design. The nodes are connected to the neighbor nodes in four directions and the nodes in the edges using wraparound links for connection.

Figure 3 illustrates the structure and connectivity of a (4 × 4) Cabinet-Level network. The hierarchical connection of the nodes grow the number of nodes in this network which can be obtained from Equation 1. Therefore, by considering $m = 2$, and by knowing the number of PEs in the Board-Level; the total number of PEs can be obtained from Equation 1 which equals to 6144 nodes.

The BM of the Cabinet-Level is a regular network that has been built based on an irregular network represented by the BM-Level and the Chip-Level. The Chip-Level has several free ports for the higher-level connection. These ports and their associated links used to connect multiple BMs to create the Cabinet-Level network, and it is divided into 2×2^q ports for the vertical connections and 2×2^q free ports for the horizontal connections, where q is inter-level connectivity and $q \in 0, 1, 2, \ldots, m$. Thus, the total number of free ports which are usable for constructing the Cabinet-Level are obtained from $4 \times 2^q = 2^{q+2}$, by choosing the minimal inter-level connectivity when $q = 0$, the number of these ports will be $4 \times 2^0 = 2^{0+2} = 4$ free ports. These ports are divided into $2 \times 2^0 = 2$ free ports for the vertical connections, and $2 \times 2^0 = 2$ free links for the horizontal connections.

D. SYSTEM-LEVEL NETWORK

The concern in recent years is to reduce the power consumption of computers and to process numerous amounts of data faster. Therefore, multi-core processors form of parallel computing became the dominant paradigm in computer architecture [15]. The System-Level of SCCN is Level-5 of hierarchy which is built based on based on SCCN topology which we have proposed in this work. Level-4 or the Cabinet-Level is the BM of this network. Expanding the network size is important to improve the performance of vital applications in various sectors such as science and engineering.

The BM of a $(2^m \times 2^m)$ System-Level network arranged in a rectangular shape with n rows and n columns. The connection between the BMs in Level-5 occurs through the free ports and their associated links which are available for higher-level interconnections in each BM. The total number of free ports in each level can be calculated by knowing the number of free ports in each immediate lower level from Equation 2.

$$fp_L = (fp_{L-1} - 2^m) \times 2^p, \tag{2}$$

where fp_L is the total number of free links in Level-L network, fp_{L-1} is the total number of the free ports in the immediate lower Level, $p = 2^m$ where $m = 2$.

The System-Level network has been created based on the hierarchical interconnection of a multiple $(2^m \times 2^m)$ Cabinet-Level in a $(2^m \times 2^m)$ 2D-torus network, where m is a positive integer. The nodes within the System-Level network are laying in 2^m rows and 2^m columns and connected through bidirectional wires to the adjacent nodes. The connection takes place in four directions north, south, east and west. Furthermore, the nodes in the edges are connected through wraparound links to minimize the distances between the nodes. To guarantee the network granularity and short distances between the nodes we will consider $(m = 2)$. Thus, a (4×4) System-Level network will be created with 16 nodes; these nodes are laying in a two-dimensional rectangular shape with 4 rows and 4 columns. Figure 4 portrays the structure of a (4×4) System-Level network. Equation 1 is used to calculate the number of PEs in this level. Therefore, the total number of nodes in Level-5 is $N_{L5} = N_{L4} \times 2^4 = N_{BM} \times 2^4 = 6144 \times 16 = 98304$ nodes. The architecture of the System-Level is defined based on the values of (m,L,q). Level-5 is a $(2^m \times 2^m)$ network and the nodes in this network arranged in 2^m rows and 2^m columns. m is a positive integer and it has a critical influence on the network structure, cost, and performance. Therefore, considering $m = 2$ is the suitable choice to create better granularity network with short distances between the nodes. L is the network level of the hierarchy, where $L = 5$ and q is the inter-level connectivity where $q \in 0, 1, 2, \ldots, m$. However, we will consider the minimal inter-level connectivity when $q = 0$. As a result, Level-5 network can be defined as SCCN (2,5,0).

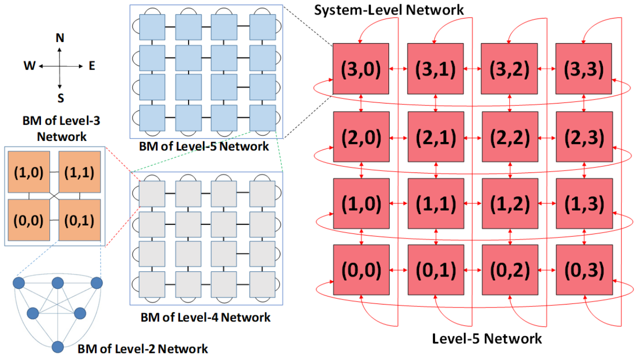

FIGURE 4. The Structural Design of System-Level Network. This is a (4 × 4) System-Level network with 16 nodes; these nodes are laying in a two-dimensional rectangular shape with 4 rows and 4 columns and are connected through bidirectional wires to the adjacent nodes. Nodes in the edges are connected through wraparound links to minimize the distances between the nodes.

III. NODE ADDRESSING OF THE SYSTEM

Knowing the exact position of each node in the network is important to track the message path to the destination node. Therefore, each node must have a unique number to define its location inside the system. The (4 × 4) Level-5 built of various lower levels connected in hierarchical design. Therefore, the address of each node in this level will compose of five parts and each part will refer to one level. Therefore, we conclude that the address of each node in L level of hierarchy composed of numerous parts calculated from the number of the lower levels which are creating this level.

The structure of the node address will be denoted in pairs of x and y values. The nodes in a (4 × 4) BM of Level-5 network are addressed in a pair of numbers based on x and y values of each node and are denoted as $A^{Level\_4} = (a_x)(a_y)$, where $0 \le (a_x)(a_y) \le \text{m} - 1$, and m = 4. Besides, the node address in a (4 × 4) Level-3 network is the BM of Level-4, and refers to the Board-Level is denoted as $A^{Level\_3} = (a_x)(a_y)$, where $0 \le (a_x)(a_y) \le \text{m} - 1$, and m = 4. Furthermore, the address of each node in the BM of Level-3 network which is referring to a (2 × 2) Chip-Level network denoted in a pair of nodes as $A^{Level\_2} = (a_x)(a_y)$, where $0 \le (a_x)(a_y) \le \text{m} - 1$, and m = 2. The nodes within Level-1 or the BM-Level network which is the BM of Level-2 representing by a unique decimal number referring to each node position in this network, these nodes represented as $A^{Level\_1} = n$, where $n \in 1, 2, 3, 4, 5, 6$. The address of each processing element in Level-L network can be obtained from Equation 3.

$$A^L = \begin{cases} n, & \text{if } L = 1, 1 \le n \le 6 \\ (n_x^L, n_y^L), & \text{if } L \ge 2 \end{cases} \tag{3}$$

In General, in Level-L network of SCCN the node address is represented by Equation 4.

$$\begin{aligned}A &= A^{L}A^{L-1}A^{L-2}A^{L-3}\ldots\ \ldots\ \ldots A^{4}A^{3}A^{2}A^{1}\\&= n_{\infty}n_{\infty-1}n_{\infty-2}n_{\infty-3}\ \ldots\ \ldots\ \ldots\ n_{3}n_{2}n_{1}n_{0}\\&= n_{2L-2}n_{2L-3}n_{2L-4}n_{2L-5}n_{2L-6}\ \ldots\ \ldots\ \ldots n_{8}\\&\quad n_{7}n_{6}n_{5}n_{4}n_{3}n_{2}n_{1}n_{0}\\&= (n_{2L-2}n_{2L-3})(n_{2L-4}n_{2L-5})n_{2L-6}\ \ldots\ \ldots\ (n_{8}n_{7})\\&\quad (n_{6}n_{5})(n_{4}n_{3})(n_{2}n_{1})n_{0}.\end{aligned}\tag{4}$$

From Equation 3, we determine the total number of digits in the address of each node in Level-5 network which equals to $n = 2L - -1$, where $L = 5$ and refers to the current level of the hierarchy, and n is the total number of digits which compose the node address.

The node is represented based on its location in each level of hierarchy by using the values of the vertical and horizontal axis of the node in each level. One group of digits represents the address of each node which include values of x and y of the node in each level of the hierarchy. For instance, $(n_{2L-4}n_{2L-5})$ is a part of a node address in a two-dimensional Level-L network, where (n_{2L-4}) is the value in x-axis and (n_{2L-5}) is the value in y-axis. Therefore, the node address in Level-5 network is denoted by $A^{L5} = (n_8n_7)(n_6n_5)(n_4n_3)(n_2n_1)n_0$, where the address is read from the left, the first pair refers to the node location in the System-Level, the second pair is the node address in the Cabinet-Level, the third pair is the node position in the Board-Level, the fourth pair refers to the node address in the Chip-Level, and finally, the last digit which is a unique decimal integer number refers to the node address in the BM-Level network.

IV. SHORTEST PATH ROUTING ALGORITHM

Routing algorithms classified as static, quasi-static and dynamic according to how adaptive they are. The static routing algorithm is simple to apply but it has some drawbacks related to traffic changes and resource failures. The routing path in the static routing is fixed and predetermined. In contrast, the dynamic routing algorithm affected by the traffic and the topological changes in choosing the suitable path but it is complex to be implemented. The shortest-path algorithm was used commonly in the quasi-static category. In this algorithm, the link distances remain fixed for a short time but it will be changed when important changes happen. In this paper, we used a simulator to determine the routing path between any two nodes in the normal condition. However, in the case of resource failures or traffic congestion, the path will be changed and increased in the worst case extra two hops as shown in Figure 5.

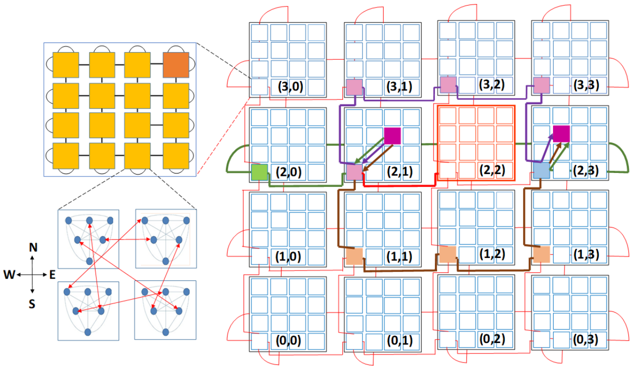

FIGURE 5. Multiple routing paths in the System-level network. The shortest-path algorithm used commonly into the quasi-static category in which the link distances remain fixed for a short time but it will be changed when important changes happen. In the case of resource failures or traffic congestion, the path will be changed and increased, in the worst case, extra two hops.

Routing a message in any level (L) of the proposed system meaning that the message will close the cycle by passing all lower levels of hierarchy starting from the BM-Level to the current level (L) and vice versa. For example, routing a message in the System-Level means that the message will pass through all levels of the hierarchy starting from the BM-Level, and then through the Chip-Level, Board-Level, Cabinet-Level and finally to the System-Level. Using wrap-around links in creating this network will make the routing mission easier by traversing only one link to send a message between the edges. The proposed routing algorithm takes into account the shortest path between any two nodes; thereby, the position of each node plays the main role in determining the routing path between any two nodes. Each BM in this system has four free ports used as gateways, thus, choosing the proper gateway depends on choosing the smallest distance between two nodes. Connecting any two nodes in any level (L) depends on determining the distance between them. For instance, in a (4 × 4) System-Level network, the nodes take numbers from 0 to 98,303, where 0 is the first node in this network and located in (0, 0) node of this System, and 98,303 is the last node and located in (3, 3) node in x and y directions of this network.

These nodes distributed in 4 rows and each row contains several nodes equal to $\frac{total\ \#\ of\ nodes}{\#\ of\ total\ nodes} = \frac{98{,}304}{4} = 24,576$ nodes. To derive the values of x and y coordinates for each node, we need to know the index number (n) of each node in the System-Level, where $0 \leq\ n \leq 98,303$. The address of each BM represented in a pair of x and y. Thus, the value of x of any node can be derived from Equation 5, and the value of y is the result of Equation 6.

$$x = \frac{N_x}{N_r}, \tag{5}$$

$$y = \frac{N_x\%N_r}{N_{BM}}, \tag{6}$$

where, N_x is the node index, while $0 \leq\ N_x\ \leq 98303$, and N_r is the total number of nodes in each row, N_{BM} is the total number of nodes in the BM of the System-Level and refers to the Cabinet-Level network.

Knowing the distance between the source and the destination nodes is essential to determine the routing path between the two nodes. Therefore, the distance between any two nodes can be measured by defining the position of each node in the System-Level by knowing the values of x and y of this node. By assuming that we have (x_1, y_1) as a source node, and (x_2, y_2) as a destination node, the distance between these two nodes can be measured from Equations 7 and 8 respectively.

$$X = |x_2 - x_1|, \tag{7}$$

$$Y = |y_2 - y_1|. \tag{8}$$

Then, the obtained results from Equations 7 and 8 will be compared if it is $\{ >, <, = \}$ the result of Equation 9.

$$\frac{1}{2} \times \frac{N_r}{N_{BM}}. \tag{9}$$

The routing decision in the System-Level will be taken based on the results of the comparison between these equations. Therefore, the packet could be routed either through the south, north, east, or west gateway. Figure 6 shows how a decision take places to route a packet between two nodes; besides, it shows the path the packet will be used to reach its destination in the System-Level network.

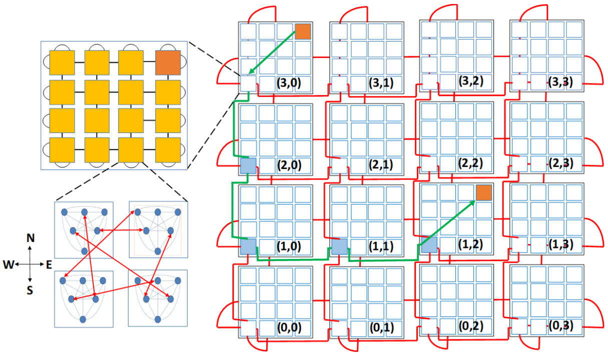

FIGURE 6. **Routing a Message in the Proposed System. Inside one node in this system, the source node needs to traverse four levels of hierarchy starting from the BM-Level and passing the Cabinet-Level to reach the System-Level of this network and then it will be routed through this level to reach another BM of the System-Level which is containing the destination node. By traversing four levels of hierarchy starting from the Cabinet-Level to the BM-Level to deliver the message to the proper destination, this operation accomplished.**

For example, from Figure 6, the routing path between two nodes illustrated in a (4 × 4) System-Level network. The nodes in this example are node number (79,870) as a source node and node number (43,005) as a destination node. The routing in this example will be from a higher value node to a lower value node. Thus, to connect these nodes, we need to determine the exact BMs in the System-Level to which each node belongs. Then we will apply the steps of determining the routing algorithm by substituting the values in the equations from 4 to 8 to measure the proper path to forward the message to its destination. Figure 7 shows how to choose the shortest path inside the Chip-Level network of this system. Besides, Figure 7 illustrates the steps of taking a decision inside the System-Level to send a message to its destination. Figures 7 and 8 are used to interpret the used simulator to determine the suitable path the message needs to follow based on the shortest path between any two nodes in the proposed system. Besides, part of the code of routing a message in this system shown in Figure 8. The routing decision in all levels which are composing the system created based on the routing mechanism in Figure 7 and Figure 8.

Simulator programs are used to calculate the routing path in the chip and the system levels as shown in Algorithm 1 and Algorithm 2 respectively. The routing algorithms and flowcharts are complex because this network is hierarchical in nature. We studied all the connection possibilities between the nodes, and determined all paths that packets should use during the connection. Therefore, we took into account all these paths during the creation of our simulation to obtain accurate results. The general explanation of this algorithm is that, when the connection between any two nodes become possible, the packet will take the shortest path to reach the destination. We have explained in the manuscript about the equations which were used in the simulator to determine the direction. These equations are Equations (3) to (8). To improve the readability of the paper, the algorithms are explained using flow chart in the paper and pseudo codes of the algorithm are included in the appendix. In this paper, we proposed a static connection between the nodes in this system by proposing this routing protocol. However, in future works, we will test dynamic protocols for this network and we will consider the connection under different circumstances such as deadlock, congestion, delay and so on. In addition, we will discuss the effect of increasing the number of paths between the nodes.

V. STATIC NETWORK PERFORMANCE PARAMETERS EVALUATION

The static network performance parameters of SCCN and its higher-levels will be evaluated in this section. A system is built from numerous levels starting from Chip-Level, Board-Level, and Cabinet-Level. Each topology has its own architecture and all of them are different from the other. This is why, maximum possible number of nodes connected by different topology is different and so, the number of nodes break off at different points. Each level in this system is evaluated by computer simulator and the results were compared to different networks. However, due to the irregularity of SCCN, it was difficult to make a fair comparison. Therefore, we compare it to hierarchical and conventional networks with almost the same number of nodes. Tables and figures are used to represent the comparison between these networks to show the attributes of each network in various levels.

Simulator: All well-known simulators were used for the evaluation of dynamic communication performance (message latency and network throughput). Dynamic communication performance is not the scope of this paper. The scope of this paper is static network performance. And for the evaluation of the static network performance, we used our own simulator to evaluate this network statically. For future works, we will use either Gem5 or bookism to evaluate the dynamic parameters of this proposed SCCN.

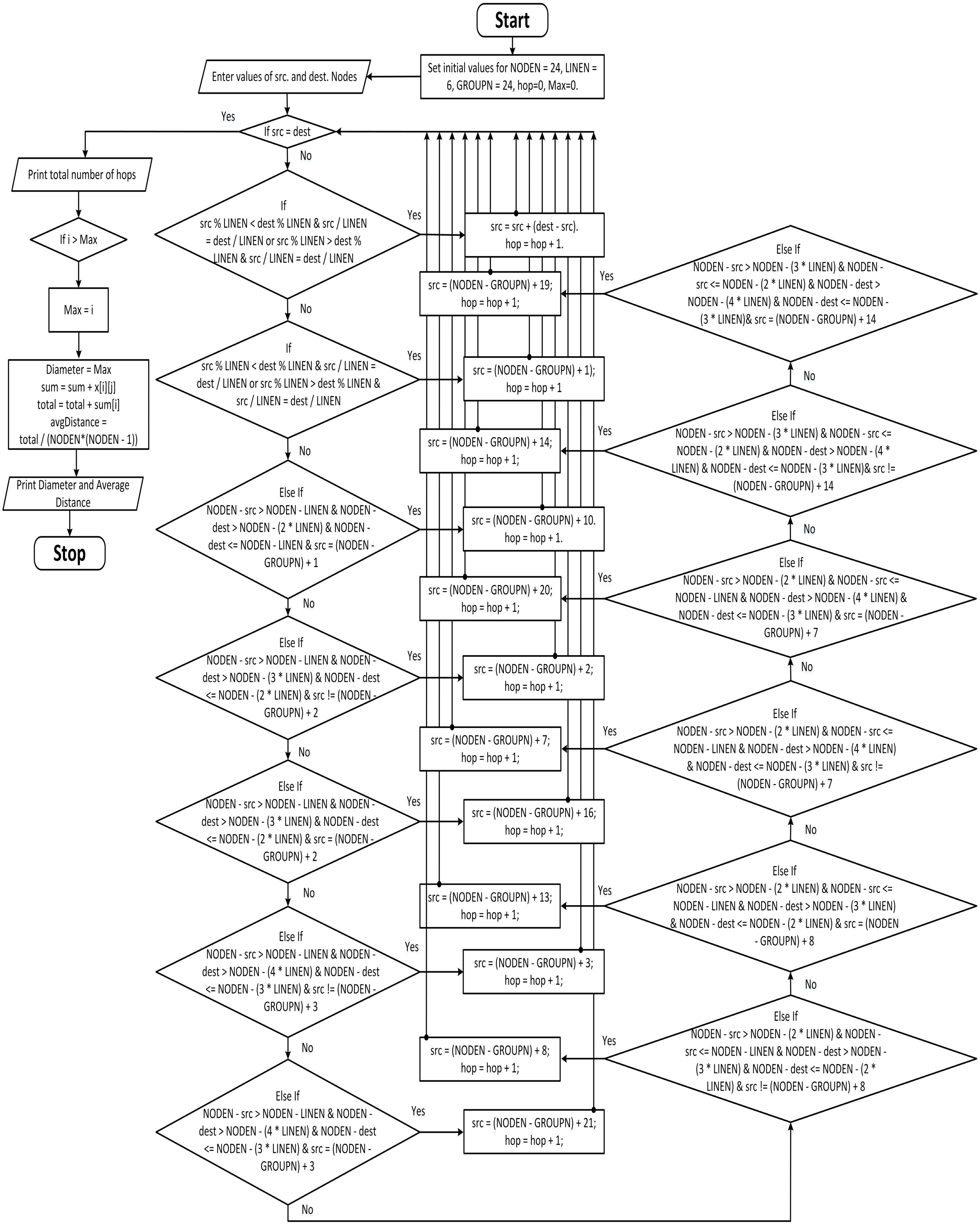
Start
Set initial values for NODEN = 24, LINEN = 6, GROUPN = 24, hop=0, Max=0.
Enter values of src. and dest. Nodes
If src = dest
Yes
No
Print total number of hops
If i > Max
Max = i
Diameter = Max
sum = sum + x[i][j]
total = total + sum[i]
avgDistance = total / (NODEN*(NODEN - 1))
Print Diameter and Average Distance
Stop
If src % LINEN < dest % LINEN & src / LINEN = dest / LINEN or src % LINEN > dest % LINEN & src / LINEN = dest / LINEN
Yes
src = src + (dest - src).
hop = hop + 1.
No
If src % LINEN < dest % LINEN & src / LINEN = dest / LINEN or src % LINEN > dest % LINEN & src / LINEN = dest / LINEN
Yes
src = (NODEN - GROUPN) + 1);
hop = hop + 1
No
Else If NODEN - src > NODEN - LINEN & NODEN - dest > NODEN - (2 * LINEN) & NODEN - dest <= NODEN - LINEN & src = (NODEN - GROUPN) + 1
Yes
src = (NODEN - GROUPN) + 10.
hop = hop + 1.
No
Else If NODEN - src > NODEN - LINEN & NODEN - dest > NODEN - (3 * LINEN) & NODEN - dest <= NODEN - (2 * LINEN) & src != (NODEN - GROUPN) + 2
Yes
src = (NODEN - GROUPN) + 2;
hop = hop + 1;
No
Else If NODEN - src > NODEN - LINEN & NODEN - dest > NODEN - (3 * LINEN) & NODEN - dest <= NODEN - (2 * LINEN) & src = (NODEN - GROUPN) + 2
Yes
src = (NODEN - GROUPN) + 16;
hop = hop + 1;
No
Else If NODEN - src > NODEN - LINEN & NODEN - dest > NODEN - (4 * LINEN) & NODEN - dest <= NODEN - (3 * LINEN) & src != (NODEN - GROUPN) + 3
Yes
src = (NODEN - GROUPN) + 3;
hop = hop + 1;
No
Else If NODEN - src > NODEN - LINEN & NODEN - dest > NODEN - (4 * LINEN) & NODEN - dest <= NODEN - (3 * LINEN) & src = (NODEN - GROUPN) + 3
Yes
src = (NODEN - GROUPN) + 21;
hop = hop + 1;
No
Else If NODEN - src > NODEN - (2 * LINEN) & NODEN - src <= NODEN - LINEN & NODEN - dest > NODEN - (3 * LINEN) & NODEN - dest <= NODEN - (2 * LINEN) & src != (NODEN - GROUPN) + 8
Yes
src = (NODEN - GROUPN) + 8;
hop = hop + 1;
No
Else If NODEN - src > NODEN - (2 * LINEN) & NODEN - src <= NODEN - LINEN & NODEN - dest > NODEN - (3 * LINEN) & NODEN - dest <= NODEN - (2 * LINEN) & src = (NODEN - GROUPN) + 8
Yes
src = (NODEN - GROUPN) + 13;
hop = hop + 1;
No
Else If NODEN - src > NODEN - (2 * LINEN) & NODEN - src <= NODEN - LINEN & NODEN - dest > NODEN - (4 * LINEN) & NODEN - dest <= NODEN - (3 * LINEN) & src != (NODEN - GROUPN) + 7
Yes
src = (NODEN - GROUPN) + 7;
hop = hop + 1;
No
Else If NODEN - src > NODEN - (2 * LINEN) & NODEN - src <= NODEN - LINEN & NODEN - dest > NODEN - (4 * LINEN) & NODEN - dest <= NODEN - (3 * LINEN) & src = (NODEN - GROUPN) + 7
Yes
src = (NODEN - GROUPN) + 20;
hop = hop + 1;
No
Else If NODEN - src > NODEN - (3 * LINEN) & NODEN - src <= NODEN - (2 * LINEN) & NODEN - dest > NODEN - (4 * LINEN) & NODEN - dest <= NODEN - (3 * LINEN)& src != (NODEN - GROUPN) + 14
Yes
src = (NODEN - GROUPN) + 14;
hop = hop + 1;
No
Else If NODEN - src > NODEN - (3 * LINEN) & NODEN - src <= NODEN - (2 * LINEN) & NODEN - dest > NODEN - (4 * LINEN) & NODEN - dest <= NODEN - (3 * LINEN)& src = (NODEN - GROUPN) + 14
Yes
src = (NODEN - GROUPN) + 19;
hop = hop + 1;

FIGURE 7. **Flowchart of Routing a Message in the Chip-Level. This figure illustrates how to choose the shortest path inside the Chip-Level network of this system. Furthermore, it demonstrates the steps of taking a decision inside the System-Level to send a message to its destination.**

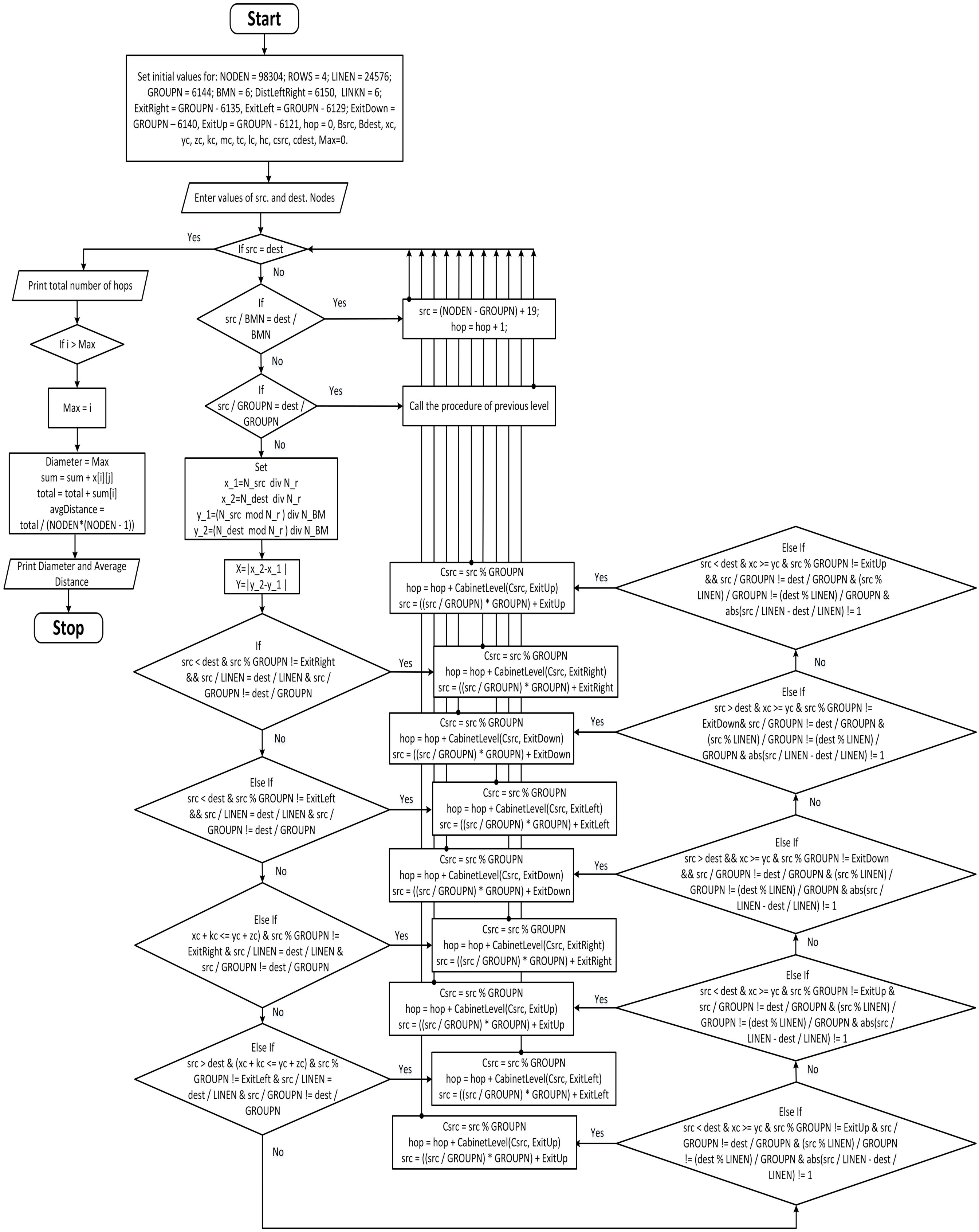
Start
Set initial values for: NODEN = 98304; ROWS = 4; LINEN = 24576; GROUPN = 6144; BMN = 6; DistLeftRight = 6150, LINKN = 6; ExitRight = GROUPN - 6135, ExitLeft = GROUPN - 6129; ExitDown = GROUPN – 6140, ExitUp = GROUPN - 6121, hop = 0, Bsrc, Bdest, xc, yc, zc, kc, mc, tc, lc, hc, csrc, cdest, Max=0.
Enter values of src. and dest. Nodes
If src = dest
Yes
No
Print total number of hops
If i > Max
Max = i
Diameter = Max
sum = sum + x[i][j]
total = total + sum[i]
avgDistance =
total / (NODEN*(NODEN - 1))
Print Diameter and Average Distance
Stop
If
src / BMN = dest /
BMN
Yes
src = (NODEN - GROUPN) + 19;
hop = hop + 1;
No
If
src / GROUPN = dest /
GROUPN
Yes
Call the procedure of previous level
No
Set
x_1=N_src div N_r
x_2=N_dest div N_r
y_1=(N_src mod N_r) div N_BM
y_2=(N_dest mod N_r) div N_BM
X=|x_2-x_1 |
Y=|y_2-y_1 |
If
src < dest & src % GROUPN != ExitRight
&& src / LINEN = dest / LINEN & src /
GROUPN != dest / GROUPN
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitRight)
src = ((src / GROUPN) * GROUPN) + ExitRight
No
Else If
src < dest & src % GROUPN != ExitLeft
&& src / LINEN = dest / LINEN & src /
GROUPN != dest / GROUPN
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitLeft)
src = ((src / GROUPN) * GROUPN) + ExitLeft
No
Else If
xc + kc <= yc + zc) & src % GROUPN !=
ExitRight & src / LINEN = dest / LINEN &
src / GROUPN != dest / GROUPN
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitRight)
src = ((src / GROUPN) * GROUPN) + ExitRight
No
Else If
src > dest & (xc + kc <= yc + zc) & src %
GROUPN != ExitLeft & src / LINEN =
dest / LINEN & src / GROUPN != dest /
GROUPN
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitLeft)
src = ((src / GROUPN) * GROUPN) + ExitLeft
No
Else If
src < dest & xc >= yc & src % GROUPN != ExitUp
&& src / GROUPN != dest / GROUPN & (src %
LINEN) / GROUPN != (dest % LINEN) / GROUPN &
abs(src / LINEN - dest / LINEN) != 1
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitUp)
src = ((src / GROUPN) * GROUPN) + ExitUp
No
Else If
src > dest & xc >= yc & src % GROUPN !=
ExitDown& src / GROUPN != dest / GROUPN &
(src % LINEN) / GROUPN != (dest % LINEN) /
GROUPN & abs(src / LINEN - dest / LINEN) != 1
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitDown)
src = ((src / GROUPN) * GROUPN) + ExitDown
No
Else If
src > dest && xc >= yc & src % GROUPN != ExitDown
&& src / GROUPN != dest / GROUPN & (src % LINEN) /
GROUPN != (dest % LINEN) / GROUPN & abs(src /
LINEN - dest / LINEN) != 1
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitDown)
src = ((src / GROUPN) * GROUPN) + ExitDown
No
Else If
src < dest & xc >= yc & src % GROUPN != ExitUp &
src / GROUPN != dest / GROUPN & (src % LINEN) /
GROUPN != (dest % LINEN) / GROUPN & abs(src /
LINEN - dest / LINEN) != 1
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitUp)
src = ((src / GROUPN) * GROUPN) + ExitUp
No
Else If
src < dest & xc >= yc & src % GROUPN != ExitUp & src /
GROUPN != dest / GROUPN & (src % LINEN) / GROUPN
!= (dest % LINEN) / GROUPN & abs(src / LINEN - dest /
LINEN) != 1
Yes
Csrc = src % GROUPN
hop = hop + CabinetLevel(Csrc, ExitUp)
src = ((src / GROUPN) * GROUPN) + ExitUp

FIGURE 8. Flowchart of Routing a Message in the System-Level. Part of the code of routing a message in this system shown. Both of the flow charts are used to interpret the used simulator to determine the suitable path the message needs to follow based on the shortest path between any two nodes in the proposed system.

Therefore, to evaluate the performance of the System-Level of SCCN consisting of 98,304 nodes, we created a simulator by using C++. In this program multiple functions are created and then called during the routing process to measure the paths taken by a message in the lower levels of this system. The functions of this simulator for system level performance evaluation are as follows:

1) **BM Function:** This function will be called when the message will be routed in the proposed basic module which is composed of six nodes.
2) **Chip Function:** This function will be called when the message will be routed in the Chip-Level network which is composed of 24 nodes.
3) **Board Function:** This function will be called when the message will be routed in the Board-Level network with 384 nodes.
4) **Cabinet Function:** This function will be called when the routing processes will be running in the Cabinet Level network with 6144 nodes.

The performance evaluation process for this large number of nodes in the system level took quite long time, almost three weeks. Therefore, the evaluation process is done by dividing this number of nodes to segments and evaluate each segment separately, finally at the end, we merged the results.

The routing protocol is the simple shortest path algorithm, but it took long time because of the permutation and combination of huge number of nodes. However, complexity of routing is low. The simulator worked with 98,304 nodes and to connect every single node to all other nodes, we divided this high number of nodes into multiple parts, and each part is run on a different computer. Finally, we merged the outcome from each computer, and we used it to simulate the final results. The work on each part of this simulator took at least two days to get the results. The computers which we used to run this simulator kept running during the duration of evaluating the system-level network.

In any network, message can be sent from any node to any other nodes, whereby the source destination pair is randomly selected. Therefore, we have considered all distinct pairs of nodes for performance evaluation. We have evaluated the hop distance of all distinct pairs of 98,304 nodes using shortest path algorithm. To calculate the hop distance parameter of these huge number of distinct pairs of nodes of a system level SCCN will take huge time. Also, the routing algorithm become complex because the proposed SCCN is a hierarchical network, not a regular network. In the simulator, we have considered all the possible ways to send the message from a source to its destination using shortest path.

A. NODE DEGREE

The input and out complexity of a network is measured by the node degree, which is the maximum node degree between all the nodes of a network and defined as the maximum number of physical links emanating from a node to be connected to the other nodes [16]. In each level of the System-Level, the BMs connected through gate nodes, chosen based on the minimum node degree. From the architecture of the BM-Level and the Chip-Level of SCCN, the minimum node degree is equal to 5 as shown in Figure 1(a) and (b). Therefore, to create a higher-level network, new physical links will be added to increase the degree to 6.

TABLE 1. Comparison between the chip-level and other networks.

Network	#	Dg	Dia	Cost	Avg.	Arc	BW	WC
2D-M	25	4	8	32	3.33	2	5	40
2D-T	25	4	5	20	2.50	4	10	50
SCCN	24	6	3	18	2.30	3	4	66
FT	16	8	6	48	5.33	4	8	48

Tables 1, 2, and 3 show the comparison between different networks and different levels of SCCN; these levels are the Chip-Level, the Board-Level, and the Cabinet-Level respectively. The node degree of SCCN in all these levels has fixed number which will maintain a fixed network cost. Figure 9 depicted the comparisons in Tables 1, 2, and 3. Besides, it depicted the comparison between the System-Level of SCCN with 98,304 nodes and multiple networks. The node degree of the hypercube network grows rapidly as the number of nodes increases. However, the other networks keep a certain degree as shown in Figure 9.

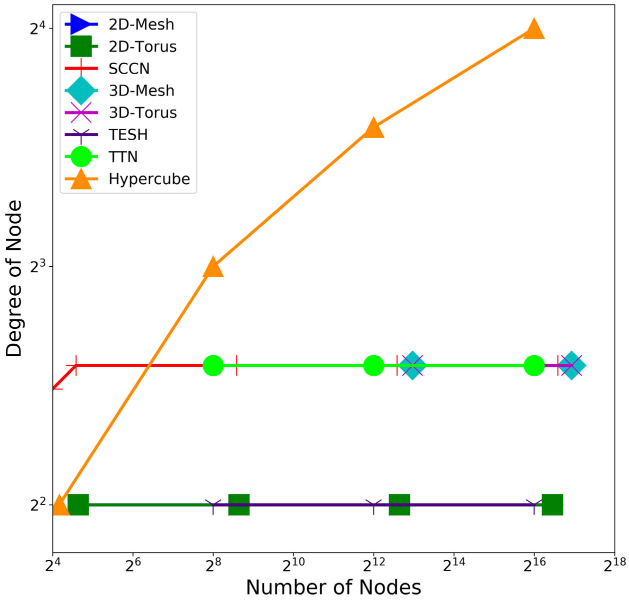

FIGURE 9. Node Degree Comparison of Various Networks. The node degree of SCCN in Chip-Level, Board-Level, and Cabinet-Level has fixed number which will maintain a fixed network cost.The node degree of the hypercube network grows rapidly as the number of nodes increases. However, the other networks keep a certain degree as shown. SCCN has node degree equal to that of TTN, 3D-mesh, 3D-Torus networks and a little bit higher than that of 2D-torus, 2D-mesh, and TESH networks.

SCCN has node degree equal to that of TTN, 3D-mesh, 3D-Torus networks and a little bit higher than that of 2D-torus, 2D-mesh, and TESH networks. Higher node degree

TABLE 2. Comparison between the board-level and other networks.

Network	# Nodes	Degree	Diameter	Average Distance	Cost	Arc Conn.	Bisection Width	WC
1D Array	384	2	383	128	766	1	1	383
1D Ring	384	2	192	96	384	2	2	384
2D Mesh	400	4	38	13.33	152	2	20	760
2D Torus	400	4	20	10	80	4	40	800
Hypercube	256	8	8	4	64	8	128	1024
STAR	384	383	2	1	766	1	192	383
TESH	256	4	21	10.47	84	2	8	416
MMN	256	4	17	9.07	68	2	8	416
TTN	256	6	15	7.44	90	4	8	576
SCCN	384	6	17	8.7	102	5	8	1088

TABLE 3. Comparison between the cabinet-level and other networks.

Network	# Nodes	Degree	Diameter	Average Distance	Cost	Arc Conn.	Bisection Width	WC
1D-Array	6144	2	6143	2048	12288	1	1	6143
Ring	6144	2	3072	1536	384	2	2	384
STAR	6144	6143	2	1	766	1	192	383
Hypercube	4096	12	12	6	144	12	2048	24576
2D-Mesh	6400	4	158	53.33	632	2	80	12640
2D-Torus	6400	4	80	40	320	4	160	12800
3D-Mesh	8000	6	57	40	342	3	400	23940
3D-Torus	8000	6	30	20	180	6	800	24000
TESH	4096	4	32	17.80	128	2	8	8192
TTN	4096	6	24	12.60	144	4	8	10240
FTTN	4096	6	26	13.14	156	4	8	8736
STTN	4096	6	25	12.50	150	4	8	8736
SCCN	6144	6	43	21.69	258	5	8	17440

increases the network complexity by increasing the number of links in the network. However, it increases the paths that packets can use to move from a source to a destination node which will control high bandwidth and mitigate congestion. SCCN has fixed and moderate degree between the networks which is illustrated in Figure 9.

B. NETWORK DIAMETER

The packet traverses several nodes in its journey from a node to another when the source and the destination nodes are not directly connected. Therefore, the distance between a source and a destination node is the total number of the traversed hops to send a message between them. In most of the interconnection networks, there are many paths between any pair of nodes and the length of the path affects the message delivery. The short path to send a message between two nodes is desirable and indicate good performance. Network diameter is defined as the maximum length among the lengths of the shortest paths between all the possible pairs of nodes in the network [17]. This means the network diameter is the worst-case distance between two nodes by choosing the shortest path, and it is proportional to the packet delay. Furthermore, it has a direct impact on the broadcasting time of a message between a source node to all other nodes. Thus, the short diameter implies using a short time to send a message.

A computer simulator is used to measure the diameter of each level in this system by taking into consideration the shortest path between the source node to all other nodes. The outcome of these simulators of each level is tabulated in Tables 1, 2, and 3. Also, Figure 10 illustrates the changes in the diameter of SCCN and multiple networks when the number of nodes in the network increase to reach the height level of the hierarchy. Each level of SCCN compared to other networks with almost the same number of nodes. Table 1 shows that Chip-Level of SCCN has the shortest networks diameter compared to 2D-mesh, 2D-torus and Fat Tree networks. Besides, as shown in Table 2, the Board-Level has the best and the shortest network diameter between all the networks in this table by taking into account the total number of nodes of each network. The network diameter of the Cabinet-Level of SCCN shows a moderate size compare to the networks in Table 3. The variances in the number of nodes as shown in Table 3 between SCCN and all other networks will minimize the difference in the diameter length between these networks.

Figure 10 depicts the network diameter comparison between multiple networks and it shows the effect of increasing the number of nodes of each network on the diameter size. SCCN diameter is better than that of 2D-mesh, 3D-mesh, and 2D-torus networks. In contrast, it is a little bit higher than that of TESH, TTN, and 3D-torus networks. However, by taking into account the number of PEs in each network the difference will be minimized. This implies the network diameter of SCCN is shorter than that of the conventional interconnection networks which used in building previous

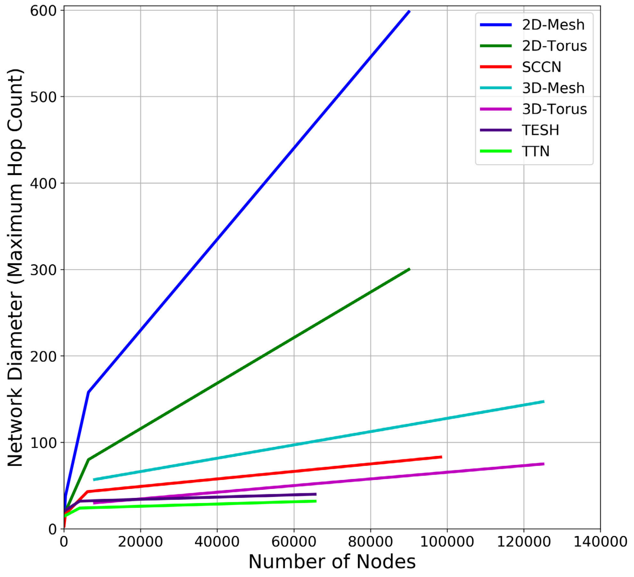

FIGURE 10. Diameter Comparison of Various Networks. The graph shows the effect of increasing the number of nodes of each network on the diameter size. SCCN diameter is better than that of 2D-mesh, 3D-mesh, and 2D-torus networks. In contrast, it is a little bit higher than that of TESH, TTN, and 3D-torus networks.

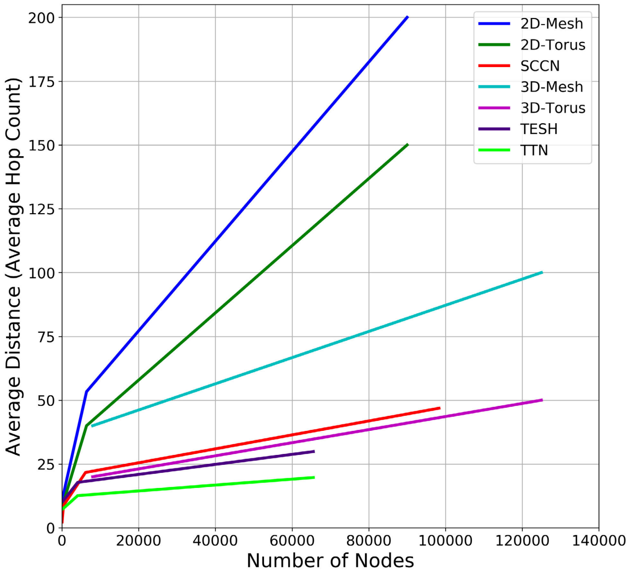

FIGURE 11. Average Distance Comparison of Various Networks. The graph displays the effect of increasing the number of nodes of each network on the average distance of this network.SCCN has better average distance than that of 2D-Mesh, 2D-Torus, 3D-Mesh, and 3D-Torus networks.

generations of MPC systems and it is suitable to make SCCN a good competitor with HINs.

C. AVERAGE DISTANCE

In multiprocessor systems, the actual performance of a network is not always reflected by network diameter because the source node is connected to many nodes during the program execution. Many short paths other than network diameter will traverse during this operation. As a result, knowing the average distance of a network is important practically to know the actual travelled distance. The network diameter is measure based on this topology. However, the average distance is measure based on the way of distributing messages between the nodes. The average distance of a network is defined as the mean distance between all distinct pairs of nodes in this network. Small average distance improves performance by decreasing the communication latency of a network [18].

The average distance of SCCN is evaluated in this section and the results are compared to various interconnection networks. The average distance of the Chip-Level, Cabinet- Level, and Board-Level was calculated by computer simulators and tabulated in Tables 1, 2, and 3 respectively. Besides, the System-Level with 98,304 nodes was assessed by a simulator and illustrated in Figure 13. SCCN has the smallest average distance compared to the other networks as shown in Tables 1 and 2. Besides, by taking into account the total number of nodes of SCCN to the total number of nodes of the other networks in Table 3, the differences in the average distance between SCCN and these networks will be trivial.

Average distance comparison of various networks is depicted in Figure 11, and it shows the effect of increasing the number of nodes of each network on average distance of this network. SCCN has better average distance than that of 2D-Mesh, 2D-Torus, 3D-Mesh, and 3D-Torus networks. However, it has a little bit higher average distance than that of TESH, and TTN networks regardless the huge difference in the total number of nodes of each network which will reduce the gap between the average distance of these networks. This implies SCCN has better performance in term of average distance than most of the popular conventional interconnection networks that are used in the industrial sector. Also, it is a strong competitor with other hierarchical interconnection networks which were introduced recently.

D. THE COST

Node degree and diameter of the interconnection network have a crucial impact on the message traffic density, inter-node distance, and fault tolerance of these networks. In multiprocessor systems, the product of (diameter × degree) is a good standard to measure the relationship between the cost and the performance of these systems. Network with high node degree has a high cost. However, long diameter decreases the network bandwidth [19]. Keeping a fixed basic configuration of a node while increasing the network size participate in providing a fixed increase in the network cost.

The cost of creating several networks is tabulated in Tables 1, 2, and 3 and illustrated in Figure 12. The cost comparison of the Chip-Level, the Board-Level, the Cabinet-Level, and the System-Level of SCCN compared to numerous networks is presented in Tables 1, 2, and 3 and Figure 12 respectively. Cost of the Chip-Level is cheap compared to the other networks as shown in Table 1. Besides, Table 2 shows that the Board-Level of SCCN has a moderate cost between other networks. However, by considering the size of SCCN and the size of other networks, we will find that the extra cost of SCCN came from the extra nodes and links which make SCCN size larger than these networks. Also, the same perception applied to the Cabinet-Level of SCCN which has extra 2,049 nodes compared to the networks in Table 3.

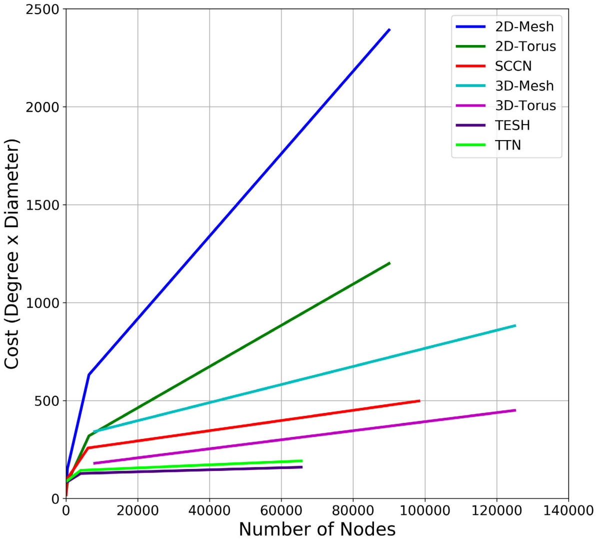

FIGURE 12. Cost Comparison of Various Networks. The figure shows the fluctuations of the cost against the increase of the number of the nodes of each network. The cost of SCCN maintains a low rate compared to 2D-mesh, 2D-torus, and 3D-mesh networks and is a little bit higher than that of TESH and TTN networks. Besides, it almost equals the cost of the 3D-Torus network.

Figure 12 shows the fluctuations of the cost against the increase of the number of the nodes of each network. The cost of SCCN maintains a low rate compared to 2D-mesh, 2D-torus, and 3D-mesh networks. Besides, it almost equals the cost of the 3D-Torus network. In contrast, Figure 12 shows that the cost rate of SCCN is a little bit higher than that of TESH and TTN networks. The highest level of SCCN has extra 32,768 nodes than the total number of nodes in the same level of TTN and TESH networks. This will participate in degrading and explaining the cost differences between these networks. Therefore, applying SCCN in MPC systems have a good impact on decreasing the total cost of creating these systems.

E. ARC CONNECTIVITY

In interconnection networks, arc connectivity is a measure for the network robustness, also, it reflects path diversity between the nodes. Network arc connectivity is the minimum number of links that must be removed to disjoint a network into two parts. Therefore, arc connectivity is always equal to or less than the node degree of the network. High arc connectivity is desirable and implies a large number of links that can be removed from a network to separate a node from its neighbors and keep the network connection; this will provide a strong and fault-tolerance network [20]. Arc Connectivity calculation of a (4 × 4) SCCN is illustrated in Figure 13.

Tables 1, 2 and 3 show arc connectivity comparison between the Chip-Level, the Board-Level, and the Cabinet-Level of SCCN respectively on one side and a collection of other networks on the other side. The Chip- Level has moderate arc connectivity compared to the networks in Table 1. In Table 2, SCCN has the highest value of arc connectivity compared to the other networks. Besides, SCCN in Table 3 shows the best result of arc connectivity compared to all networks except 3D-Torus. The effect of increasing the number of nodes of multiple conventional and hierarchical interconnection networks on arc connectivity of these networks are depicted in Figure 14.

SCCN arc connectivity is higher than that of 1D-Array, Ring, 2D-mesh, 2D-torus, 3D-mesh, TESH, and TTN. However, it is a little bit lower than that of 3D-torus network. High arc connectivity of SCCN compared to the other networks makes it more fault tolerance and more robust than these networks. This qualifies SCCN to be a good choice in terms of arc connectivity over all these networks for MPC systems.

F. BISECTION WIDTH (BW)

Bisection width (BW) is an important attribute to measure the performance of the interconnection networks and it has a crucial impact on the cost of VLSI layout of these networks. Furthermore, BW is vital to measure the traffic volume that can be controlled by a network and is useful in assessing the area required from the VLSI implementation of this network. BW of a network defined as the minimum number of wires that must be cut to separate the network into two halves [21]. The higher Levels of SCCN are created based on the hierarchical interconnection of the lower-level networks connected in a 2D-torus network. Since the BW of 2D-torus network with N number of nodes is $2\sqrt{N}$, thus, we can derive BW of a $(2^m \times 2^m)$ SCCN network from Equation 10.

$$BW(SCCN) = 2\sqrt{2^m \times 2^m}. \tag{10}$$

Here we consider SCCN is a (4 × 4) network and by substituting that in Equation (9) we will get BW of the higher levels of SCCN starting from the Board-Level which is equal to 8. This indicates that to divide these Levels to equal parts, we must remove 8 links from each network. Solving many problems by separating input data into two equal parts is beneficial. Data in each part will be processed separately and then the results will be merged to provide a final solution to the problem. Small BW is desirable for efficient VLSI realization; however, it means low bandwidth which will affect in slowing the merge of the final results of the two parts. In contrast, large BW indicates a network with fast data exchange leading to a high degree of fault tolerance. That means large BW is more desirable but it needs a large layout area of VLSI implementation. As a result, we concluded that the best BW of a network is the moderate one which means the network has fast data exchange and effective VLSI implementation. Table 1 shows that BW of the Chip-Level of SCCN is extremely lower than that of 2D-torus and Fat Tree networks and almost equal to that of 2D-mesh.

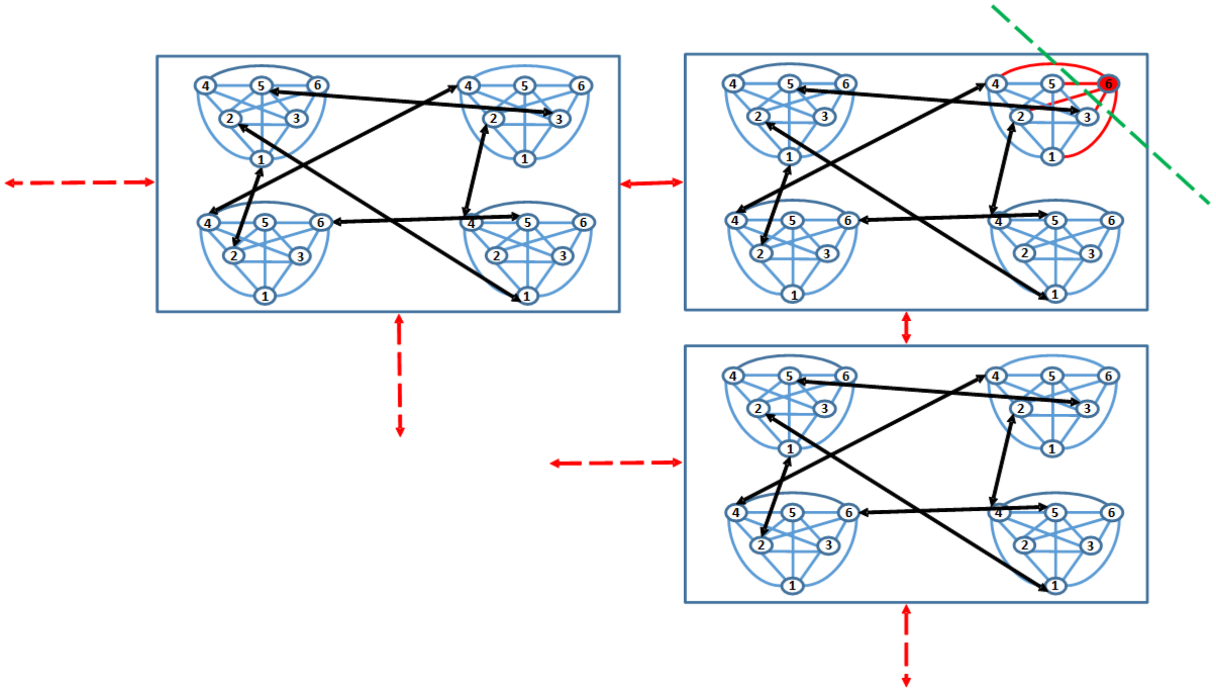

FIGURE 13. Arc Connectivity calculation of the proposed network. Removing 5 links disconnects the network into two disjoint parts. Node 6 is removed by cutting the 5 wires which are connecting it to the other nodes. That is the minimum number of links that can be removed to disconnect a node within this network which implies that the arc connectivity of Level-3 network of SCCN is 5.

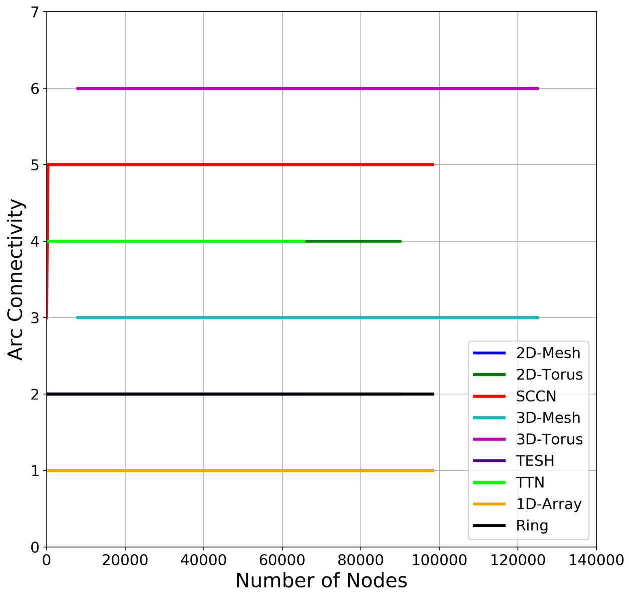

FIGURE 14. Arc connectivity Comparison of Different networks. The graph depicts The effect of increasing number of nodes of multiple conventional and hierarchical interconnection networks on arc connectivity of these networks. SCCN arc connectivity is higher than that of 1D-Array, Ring, 2D-mesh, 2D-torus, 3D-mesh, TESH, and TTN nonetheless it is a little bit lower than that of 3D-torus network.

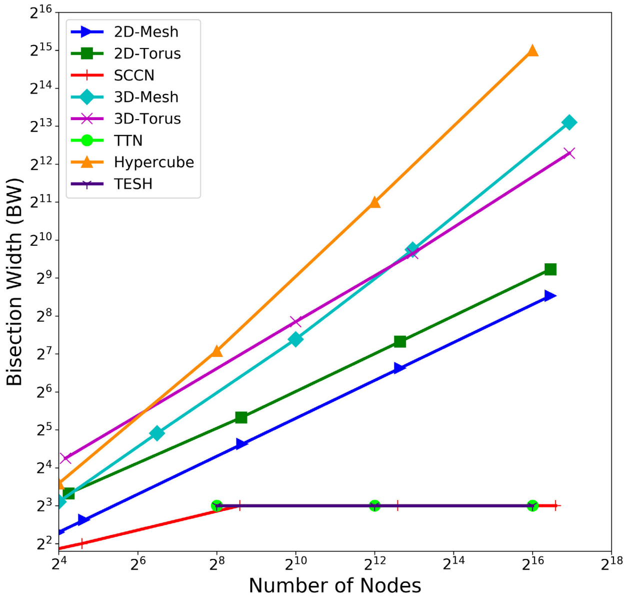

FIGURE 15. Bisection width comparison of different networks. SCCN maintains the same value of BW starting from the Board-Level to the System-Level network.BW of SCCN is lower than that of the hypercube, 3D-mesh, 3D-torus, 2D-mesh, and 2D-torus, and practically higher than that of Array, and Ring networks.

Tables 2 and 3 exposed that the Board-Level and the Cabinet- Level of SCCN has a fixed value of BW and it is moderate compared to the other networks in these comparisons. Increasing the size of SCCN by increasing the total number of nodes does not affect BW of this network. Figure 15 shows that SCCN maintains the same value of BW starting from the Board-Level to the System-Level network. Besides, it shows BW of SCCN is lower than that of the hypercube, 3D-mesh, 3D-torus, 2D-mesh, and 2D-torus, and practically higher than that of Array, and Ring networks. This implies that SCCN poses a moderate bisection width which will grant SCCN a fast data exchange and effective VLSI implementation. On the other hand, BW of Hypercube, 3D-torus, and 3D-mesh is extremely high which implies the implementation of these networks needs a large layout area of VLSI which is undesirable in MPC systems. Also, networks such as Ring, and Array have too small BW which implies a slow data exchange. Consequently, SCCN is suitable compared to these networks for MPC systems in term of BW.

G. WIRING COMPLEXITY (WC)

Wiring complexity (WC) of an interconnection network related to the number of physical links and is defined as the total number of wires needed to connect each node to the other nodes in this network. Network complexity increases as the number of wires increase which has a significant influence on the cost of the network. High wiring complexity increase the cost of the network; however, low wiring complexity indicates low cost. In this section, we will calculate the wiring complexity of SCCN in multiple levels of hierarchy. Therefore, from the lower levels which are composing this system, we will derive a formula to measure WC of the highest level with 98,304 nodes. Equation 11 is used to measure WC for the BM-Level.

$$WC_{BM} = \left[\frac{N}{2} + (N \times 2)\right], \qquad (11)$$

N is the PEs in the proposed topology which equals to 6. Besides, Equation 12 to measure WC of the Chip-Level.

$$WC_{L2} = \left[\left[\frac{N}{2} + (N \times 2)\right] \times 4\right] + 6, \qquad (12)$$

N is the PEs in the proposed topology which equals to 6. Equation 13 used to measure WC of the Board-Level.

$$WC_{L3} = ((N_x \times N_y) \times (((\frac{N_{BM}}{2} + (N_{BM} \times 2) \times 4) + 6)) \\ +(2 \times (N_x \times N_y)), \qquad (13)$$

where N_{BM} the entire number of nodes in the BM, and $\boldsymbol{N_x}$ is the overall number of nodes in x -direction of the Board Level and $\boldsymbol{N_y}$ is the overall number of nodes in y -direction of this network. Equation 14 is used to calculate the wiring complexity of the Cabinet-Level of SCCN.

$$\begin{aligned} WC_{L4} = {} & ((N_x \times N_y) \times ((N_{xl3} \times N_{yl3}) \\ & \times(((\frac{N_{BM}}{2} + (N_{BM} \times 2) \times 4) + 6) \\ & +(2 \times (N_{xl3} \times N_{yl3})))) + (2 \times (N_x \times N_y)). \qquad (14) \end{aligned}$$

Here, N_{BM} is the PEs in the BM-level, N_{xl3} is the overall number of nodes in x -direction of the Board-Level, N_{yl3} is the entire number of nodes in y -direction of the Board-Level. N_x is the PEs in x -direction of the Cabinet-Level, and N_y is the PEs in y -direction of the Cabinet-Level.

The System-Level is composed of multiple cabinet levels connected in a 2D-torus network in hierarchical design. Therefore, WC of $\left(N_x \times N_y\right)$ System-Level will be a combination of the wiring complexity of all the previous levels and it can be measured from Equation 15, N_x is the PEs in x -axis of this network and N_y is the PEs in y -direction of the System-Level.

$$\begin{aligned} & WC_{L5} \\ & = ((N_x \times N_y) \times (((N_{xl4} \times N_{yl4}) \times ((N_{xl3} \times N_{yl3}) \\ & \times(((\frac{N_{BM}}{2} + (N_{BM} \times 2) \times 4) + 6) + (2 \times (N_{xl3} \\ & \times N_{yl3})))) + (2 \times (N_{xl4} \times N_{yl4})))) + (2 \times (N_x \times N_y)). \\ & \qquad (15) \end{aligned}$$

Here, N_{xl4} represent the entire number of nodes in x -axis of the Cabinet-Level, N_{yl4} is the PEs in y -axis of the Cabinet Level, N_{xl3} the PEs in x -axis of the Board-Level, N_{yl3} is the entire number of nodes in y -axis of the Board-Level, and N_{BM} is the overall number of nodes in the BM of this network which equal to 6. By substituting these values in Equation 14 we will get the wiring complexity of the System-Level of SCCN.

Building a system based on a completely connected topology will increase the physical links in internal network. The nodes in the BM of SCCN are connected completely. However, in the higher levels, to reduce the complexity of these levels, we connected each lower level in a 2D-torus network. In Table 1, WC of the Chip-Level is higher than the other networks. Besides, Table 2 tabulates wiring complexity comparison between the Board-Level of SCCN and multiple networks and it shows SCCN recorded the highest number of wires compared to these networks. Table 3 shows that WC of the Cabinet-Level is lower than that of 3D-Mesh and 3D-torus networks and higher than that of the other networks that were included in this comparison.

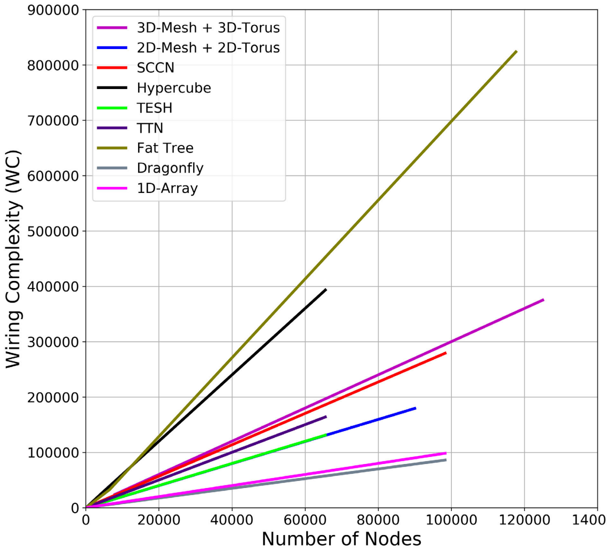

FIGURE 16. Wiring Complexity Comparison of Different networks. The graph compare the increase of the network size to the increase of its wiring complexity.Wiring complexity of SCCN is lower than that of 3D-mesh, 3D-torus, Hypercube, and Fat Tree networks but it is higher than that of TTN, TESH, 2D-mesh, 2D-torus and Dragonfly networks.

Figure 16 compares the increase of the network size to the increase of its wiring complexity. SCCN has a moderate wiring complexity between all these networks in almost all levels. Wiring complexity of SCCN is lower than that of 3D-mesh, 3D-torus, Hypercube, and Fat Tree networks. However, it is higher than that of TTN, TESH, 2D-mesh, 2D-torus and Dragonfly networks. A little bit higher wiring complexity of SCCN achieved a network with good performance compared to many conventional and hierarchical interconnection networks. This could qualify SCCN to be a better choice in building future generations of MPC systems.

VI. SOME GENERALIZATIONS

A system is proposed based on SCCN; this network started with 6 nodes and expanded to 98,304 nodes. The total number of nodes in this system is higher than that of the world's most powerful supercomputer which is called Summit. It composes of 4,356 nodes [19]. Moreover, the proposed system has number of nodes higher than that of Sunway Taihulight supercomputer which has 40,960 nodes and considered as the second powerful supercomputer in the world [22]–[24].

The proposed system composed of multiple stages including the Chip-Level stage with 24 nodes, the Board-Level stage with 384 nodes, and the Cabinet-Level stage with 6144 nodes. All these levels evaluated and compared to different networks with almost the same number of nodes. SCCN showed better results than the other networks in most of these comparisons. For instance, SCCN with 24 nodes has a better diameter and average distance than 2D-mesh, and 2D-torus. Furthermore, SCCN with 384 nodes has also a better diameter and average distance than 2D-mesh, 2D-torus, TESH, MMN, and TTN. Besides, SCCN with 6144 nodes has a better diameter and average distance than 2D-mesh. 2D-torus, and 3D-mesh networks.

Furthermore, SCCN showed good results in many aspects such as cost, arc connectivity, bisection width, and wiring complexity than multiple networks which we have included in these comparisons. The size of SCCN expanded better than all other networks as clarified in this paper which participates in creating a system with a huge number of nodes. As a result, SCCN could be a good choice over many conventional and hierarchical interconnection networks for MPC systems.

VII. LIMITATIONS AND FUTURE WORK

The limitation of this paper is to propose a topology for MPC systems. This topology used as a BM to build the higher-level networks by interconnecting multiple levels networks in a hierarchical fashion to build an MPC system. The static networks performance of each level is evaluated by a computer simulator, built by C++, dedicated for evaluating the network statically under the normal conditions by using the short path routing protocol. Finally, we compared the obtained results with that of different networks.

For future work, we will assess the dynamic communication performance of each level in the proposed system, and propose 3-dimensional design of SCCN (3D-SCCN). A hierarchical interconnection network (HIN) is a plausible alternative way to interconnect millions of nodes for the next generation MPC system. However, no HIN is a plausible alternative option for the industry community yet. Thus, HIN is a promising research area for the MPC system. Dynamic performance using dynamic traffic patterns for a network topology is highly desired, and it is tiring and time-consuming for a HIN consisting of a huge number of nodes. The very

Algorithm 1 Routing Protocol of a Message in the Chip-Level Network

```
Define Network Nodes:
input: ($N = 24$, $G_n = 24$, $M_n = 6$, $s_i$, $d_j$, hop, i, j)
Optimize Network State:
output: (Sum, Diameter, Average-Distance)
Initialization:
(hop = 0, Sum = 0)
Begin:
for i = 0 to N-1 do
  for j = 0 to N-1 do
    While($s_i \neq d_j$) do
      if ($s_i$ mod $M_n < d_j$ mod $M_n$ and $s_i$ div $M_n = d_j$ div $M_n$ or $s_i$ mod $M_n > d_i$ mod $M_n$ and $s_i$ div $M_n = d_j$ div $M_n$) then $s_i = s_i + (d_j - s_i)$; hop = hop + 1;
      else if ($N - s_i > N - M_n$ and $N - d_j > N - (2 * M_n)$ and $N - d_j <= N - M_n$ and $s_i \neq (N - G_n) + 1$ or $N - s_i > N-(2 * M_n)$ and $N - s_i <= N - M_n$ and $N - d_j > N - M_n$ and $s_i = (N - Gn) + 10$) then $s_i = (N - G_n) + 1$; hop = hop + 1; //The connection between Group A and B
      else if ($N - s_i > N - M_n$ and $N - d_i > N - (2 * M_n)$ and $N - d_i <= N - M_n$ and $s_i = (N - G_n) + 1$ or $N - s_i > N-(2 * M_n)$ and $N - s_i <= N - M_n$ and $N - d_j > N - M_n$ and $s_i \neq (N - G_n) + 10$) then $s_i = (N - G_n) + 10$; hop = hop + 1; // Delivering the message from A to B
      else if ($N - s_i > N - M_n$ and $N - d_j > N - (3 * M_n)$ and $N - d_j <= N - (2 * M_n)$ and $s_i \neq (N - G_n) + 2$ or $N - s_i > N-(3 * M_n)$ and $N - s_i <= N -(2 * M_n)$ and $N - d_j > N - M_n$ and $s_i = (N - G_n) + 16$) then $s_i = (N - G_n) + 2$; hop = hop + 1; //The connection between Group A and C
      else if ($N - s_i > N - M_n$ and $N - d_j > N - (3 * M_n)$ and $N - d_j <= N - (2 * M_n)$ and $s_i = (N - G_n) + 2$ or $N - s_i > N-(3 * M_n)$ and $N - s_i <= N - (2 * M_n)$ and $N - d_j > N - M_n$ and $s_i \neq (N - G_n) + 16$) then $s_i = (N - G_n) + 16$; hop = hop + 1; // Delivering the message from A to C
      else if ($N - s_i > N - M_n$ and $N - d_j > N - (4 * M_n)$ and $N - d_j <= N - (3 * M_n)$ and $s_i \neq (N - G_n) + 3$ or $N - s_i > N-(4 * M_n)$ and $N - s_i <= N -(3 * M_n)$ and $N - d_j > N - M_n$ and $s_i = (N - G_n) + 21$) then $s_i = (N - G_n) + 3$; hop = hop + 1; //The connection between Group A and D
      else if ($N - s_i > N - M_n$ and $N - d_j > N - (4 * M_n)$ and $N - d_j <= N - (3 * M_n)$ and $s_i = (N - G_n) + 3$ or $N - s_i > N-(4 * M_n)$ and $N - s_i <= N - (3 * M_n)$ and $N - d_j > N - M_n$ and $s_i \neq (N - G_n) + 21$) then $s_i = (N - G_n) + 21$; hop = hop + 1; // Delivering the message from A to D
      else if ($N - s_i > N -(2 * M_n)$ and $N - s_i <= N - M_n$ and $N - d_j > N - (3 * M_n)$ and $N - d_j <= N - (2 * M_n)$ and $s_i \neq (N - G_n) + 8$ or $N - s_i > N-(3 * M_n)$ and $N - s_i <= N -(2 * M_n)$ and $N - d_j > N -(2 * M_n)$ and $N - d_j <= N - M_n$ and $s_i = (N - G_n) + 13$) then $s_i = (N - G_n) + 8$; hop = hop + 1; //The connection between Group B and C
      else if ($N - s_i > N -(2 * M_n)$ and $N - s_i <= N - M_n$ and $N - d_j > N - (3 * M_n)$ and $N - d_i <= N - (2 * M_n)$ and $s_i = (N - G_n) + 8$ or $N - s_i > N-(3 * M_n)$ and $N - s_i <= N -(2 * M_n)$ and $N - d_j > N -(2 * M_n)$ and $N - d_j < = N - M_n$ and
```

Algorithm 1 *(Continued.)* Routing Protocol of a Message in the Chip-Level Network

```
$s_i \neq (N – G_n) + 13)$ then $s_i = (N – G_n) + 13$; hop = hop + 1; //Delivering the message from B to C
        else if $(N – s_i > N -(2 * M_n)$ and $N – s_i <= N - M_n$ and $N – d_j > N - (4 * M_n)$ and $N – d_j <= N - (3 * M_n)$ and $s_i \neq (N – G_n) + 7$ or $N – s_i > N-(4 * M_n)$ and $N - s_i <= N –(3 * M_n)$ and $N – d_j > N –(2 * M_n)$ and $N – d_j <= N - M_n$ and $s_i = (N – G_n) + 20)$ then $s_i = (N – G_n) + 7$; hop = hop + 1; //The connection between Group B and D
        else if $(N – s_i > N -(2 * M_n)$ and $N – s_i <= N - M_n$ and $N – d_j > N - (4 * M_n)$ and $N – d_j <= N - (3 * M_n)$ and $s_i = (N – G_n) +7$ or $N – s_i > N-(4 * M_n)$ and $N - s_i <= N –(3 * M_n)$ and $N – d_j > N –(2 * M_n)$ and $N – d_j <= N - M_n$ and $s_i \neq (N – G_n) + 20)$ then $s_i = (N – G_n) + 20$; hop = hop + 1; //Delivering the message from B to D
        else if $(N – s_i > N -(3 * M_n)$ and $N – s_i <= N - (2 * M_n)$ and $N – d_j > N - (4 * M_n)$ and $N – d_j <= N - (3 * M_n)$ and $s_i \neq (N – G_n) + 14$ or $N – s_i > N-(4 * M_n)$ and $N - s_i <= N –(3 * M_n)$ and $N – d_j > N –(3 * M_n)$ and $N – d_j <= N - (2 * M_n)$ and $s_i = (N – G_n) + 19)$ then $s_i = (N – G_n) + 14$; hop = hop + 1; //The connection between Group C and D
        else if $(N – s_i > N -(3 * M_n)$ and $N – s_i <= N - (2 * M_n)$ and $N – d_j > N - (4 * M_n)$ and $N – d_j <= N - (3 * M_n)$ and $s_i = (N – G_n) + 14$ or $N – s_i > N-(4 * M_n)$ and $N - s_i <= N –(3 * M_n)$ and $N – d_j > N –(3 * M_n)$ and $N – d_j <= N - (2 * M_n)$ and $s_i \neq (N – G_n) + 19)$ then $s_i = (N – G_n) + 19$; hop = hop + 1; //Delivering the message from C to D
            End if
      End for
End for
```

first assessment of topology is the static network performance which predicts the nature of the dynamic performance. The first step of research on SCCN is the evaluation of static network performance and that is the scope of this paper. And good static network performance (low diameter and average distance) predicts good dynamic performance (low latency and high throughput) under various dynamic traffic patterns. In our next step of research, we will evaluate the dynamic performance under uniform, hot-spot, and non-uniform traffic patterns using computer simulation.

VIII. CONCLUSION

In this paper, we proposed a massively parallel computer system, built based on a proposed hierarchical interconnection network topology called Shifted Completely Connected Network (SCCN). This topology is composed of six nodes connected completely. The lower levels placed in a hierarchical design to create the higher levels of the proposed system and it is discussed in details in this paper. The architecture of the Chip-Level, the Board-Level, the Cabinet-Level, and the System-Level of a proposed MPC system have been discussed. Besides, the node location in each level of this system is discussed in details. The shortest path routing protocol proposed to connect any two nodes in this system. Static network performance parameters including node degree, diameter, average distance, cost, arc connectivity, bisection width, and wiring complexity of each level of SCCN evaluated and compared to various networks. SCCN showed better results in many aspects compared to these networks.

Algorithm 2 Routing Protocol of a Message in the System-Level Network

```
Define Network Nodes:
Input: (N = 98304, $G_n$ = 6144, $L_n$ = 24576, $R_s$ = 4, $D_{lr}$ = 6150, src, dest, x, y, z, k, m, t, l, h, bsrc, bdest, ROWS = 4)
Define the Gate Nodes in 4 directions (Right, Left, Down, Up):
Input: ($E_r = G_n - 6135$, $E_l = G_n - 6129$, $E_d = G_n - 6140$, $E_u = G_n - 6121$)
Optimize Network State: Output: (Sum, Diameter, Average, Distance)
Initialization: (hop = 0, Sum = 0) Function BL // function to connect the nodes in the lower level networks //Connecting two nodes in system level when Source node (src) > Dest. node (dest) and src mod no. of nodes in each line ($L_n$) < dest mod $L_n$
For i = 0 to N-1 do
    For j = 0 to N-1 do
        While (src $\neq$ dest) do
            if absolute (dest div ($L_n$) – src div ($L_n$)) = (($L_n$) div ($G_n$)) div 2 and absolute ((dest mod ($L_n$)) div ($G_n$) – (src mod ($L_n$)) div ($G_n$)) < (($L_n$) div ($G_n$)) div 2 and src div ($L_n$) $\neq$ dest div ($L_n$) and src > dest and src mod ($L_n$) < dest mod ($L_n$) and (src mod ($L_n$)) div ($G_n$)) $\neq$ (dest mod ($L_n$)) div ($G_n$))
            Begin:
                bsrc = src mod $G_n$, x = BL (bsrc, $E_r$)
                bsrc = src mod $G_n$, y = BL (bsrc, $E_d$)
                bsrc = src mod $G_n$, m = BL (bsrc, $E_u$)
                bdest = dest mod $G_n$, z = BL (bdest, $E_l$)
                bdest = dest mod $G_n$, k = BL (bdest, $E_u$)
                bdest = dest mod $G_n$, t = BL (bdest, $E_d$)
            End if
                if (src > dest and (y + z <= x + k) and (y + z <= x + t) and (y + z <= m + z) and (y + z <= m + t) and src mod $G_n \neq E_d$ and src div $G_n \neq$ dest div $G_n$ and (src mod $L_n$) div $G_n \neq$ (dest mod $L_n$)) div $G_n$ or src > dest and (y + k <= x + k) and (y + k <= x + t) and (y + k <= m + z) and (y + k <= m + t) and src mod $G_n \neq E_d$ and src div $G_n \neq$ dest div $G_n$ and (src mod $L_n$)) div $G_n \neq$ (dest mod $L_n$))div $G_n$)
                Begin:
                    bsrc = src mod $G_n$
                    hop = hop + BL (bsrc,$E_d$)
                    src = ((src div $G_n$) * $G_n$) + $E_d$
                    End if
                    if (src > dest and src mod $G_n = E_d$ and (src mod $L_n$)div $G_n \neq$ (dest mod Ln)div $G_n$)
                    Begin:
                        src = src – $L_n$ + 19
                        hop = hop + 1
                    End if
```

Algorithm 2 *(Continued.)* Routing Protocol of a Message in the System-Level Network

```
        if (src > dest and ($x + k <= y + z$) and ($x + k <= y + k$) and ($x + k <= m + z$) and ($x + k <= m + t$) and src mod $G_n \neq E_r$ and src div $G_n \neq$ dest div $G_n$ and (src mod $L_n$) div $G_n \neq$ (dest mod $L_n$) div $G_n$ or src > dest and ($x + t <= y + z$) and ($x + t <= y + k$) and ($x + t <= m + z$) and ($x + t <= m + t$) and src mod $G_n \neq E_r$ and src div $G_n \neq$ dest div $G_n$ and (src mod $L_n$) div $G_n \neq$ (dest mod $L_n$) div $G_n$)
    Begin:
        bsrc = src mod $G_n$
        hop = hop + BL(bsrc, $E_r$)
        src = ((src div $G_n$) * $G_n$) + $E_r$
      End if
        if (src > dest and src mod $G_n = E_r$ and (src mod $L_n$) div $G_n \neq$ (dest mod $L_n$) div $G_n$)
    Begin:
        src = src + $D_{lr}$
        hop = hop + 1
      End if
        if (src > dest and ($m + z <= x + k$) and ($m + z <= x + t$) and ($m + z <= y + z$) and ($m + z <= y + k$) and src mod $G_n \neq$ Eu and src div $G_n \neq$ dest div $G_n$ and (src mod $L_n$) div $G_n \neq$ (dest mod $L_n$) div $G_n$ or src > dest and ($m + t <= x + k$) and ($m + t <= x + t$) and ($m + t <= y + z$) and ($m + t <= y + k$) and src mod $G_n \neq E_u$ and src div $G_n \neq$ dest div $G_n$ and (src mod $L_n$) div $G_n \neq$ (dest mod $L_n$) div $G_n$)
    Begin:
        bsrc = src mod $G_n$
        hop = hop + BL(bsrc, $E_u$)
        src = ((src div $G_n$) * $G_n$) + $E_u$
      End if
        if (src > dest and src mod $G_n$) = $E_u$ and src div $L_n \neq$ ROWS - 1 and (src mod $L_n$) div $G_n$) $\neq$ (dest mod $L_n$) div $G_n$))
    Begin:
        src = src + $L_n$ - 19
        hop = hop + 1
      End if
        if (src > dest and src mod $G_n$) = $E_u$ and src div $L_n$ = ROWS - 1 and (src mod $L_n$) div $G_n$) $\neq$ (dest mod $L_n$) div $G_n$))
    Begin:
        src = (src mod $L_n$) - 19
        hop = hop + 1
       End if
  End for
End for
```

APPENDIX. ROUTING PATH SIMULATION

The following algorithms implements the calculation of the routing path in the chip and the system levels:

REFERENCES

[1] B. Barney, "Introduction to parallel computing," *Lawrence Livermore Nat. Lab.*, vol. 6, no. 13, p. 10, Dec. 2010.

[2] D. Sarkar, "Cost and time-cost effectiveness of multiprocessing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 6, pp. 704–712, Jun. 1993.

[3] M. J. Forsell, "Architectural differences of efficient sequential and parallel computers," *J. Syst. Archit.*, vol. 47, no. 13, pp. 1017–1041, Jul. 2002.

[4] Z. Wang and J. Crowcroft, "Analysis of shortest-path routing algorithms in a dynamic network environment," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 22, no. 2, pp. 63–71, Apr. 1992.

[5] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. Int. Symp. Comput. Archit.*, Jun. 2008, pp. 77–88.

[6] M. N. Ali, M. H. Rahman, R. M. Nor, and T. M. B. T. Sembok, "A high radix hierarchical interconnection network for network-on-chip," in *Recent Advances in Information and Communication Technology*. Khon Kaen, Thailand: Springer, 2016, pp. 245–254.

[7] A. Agarwal, "Limits on interconnection network performance," *IEEE Trans. Parallel Distrib. Syst.*, vol. 2, no. 4, pp. 398–412, Oct. 1991.

[8] M. N. M. Ali, M. M. H. Rahman, R. M. Nor, D. K. Behera, T. M. T. Sembok, Y. Miura, and Y. Inoguchi, "SCCN: A time-effective hierarchical interconnection network for network-on-chip," *Mobile Netw. Appl.*, vol. 24, no. 4, pp. 1255–1264, Aug. 2019.

[9] M. Abd-El-Barr and T. F. Al-Somani, "Topological properties of hierarchical interconnection networks: A review and comparison," *J. Electr. Comput. Eng.*, vol. 2011, pp. 1–12, Jan. 2011.

[10] V. F. Pavlidis and E. G. Friedman, "3-D topologies for networks-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1081–1090, Oct. 2007.

[11] H. Amano, "Tutorial: Introduction to interconnection networks from system area network to network on chips," in *Proc. 1st Int. Symp. Comput. Netw.*, Dec. 2013, pp. 15–16.

[12] A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. B.-D. Rubin, F. Akopyan, E. McQuinn, W. P. Risk, and D. S. Modha, "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–10.

[13] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, and B. Brezzo, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[14] M. N. Ali, M. H. Rahman, A. A. Ibrahim, D. K. Behera, and Y. Inoguchi, "The connectivity and the static-cost-effective analysis of a shifted completely connected network," *Int. J. Comput. Intell. Stud.*, vol. 8, nos. 1–2, pp. 158–175, 2019.

[15] G. S. Almasi and A. Gottlieb, *Highly Parallel Computing*. Redwood City, CA, USA: Benjamin-Cummings, 1988.

[16] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2002, pp. 80–91.

[17] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, "The diameter of opportunistic mobile networks," in *Proc. ACM CoNEXT Conf. (CoNEXT)*, 2007, pp. 1–12.

[18] M. A. Bender, D. P. Bunde, E. D. Demaine, S. P. Fekete, V. J. Leung, H. Meijer, and C. A. Phillips, "Communication-aware processor allocation for supercomputers: Finding point sets of small average distance," *Algorithmica*, vol. 50, no. 2, pp. 279–298, Feb. 2008.

[19] M. Besta and T. Hoefler, "Slim fly: A cost effective low-diameter network topology," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2014, pp. 348–359.

[20] L. Volkmann, "Restricted arc-connectivity of digraphs," *Inf. Process. Lett.*, vol. 103, no. 6, pp. 234–239, Sep. 2007.

[21] J. Hines, "Stepping up to summit," *Comput. Sci. Eng.*, vol. 20, no. 2, pp. 78–82, Mar. 2018.

[22] (2019). *November 2019 | Top500*. [Online]. Available: https://top500.org/lists/top500/2019/11/

[23] H. Fu, J. Liao, J. Yang, L. Wang, Z. Song, X. Huang, C. Yang, W. Xue, F. Liu, F. Qiao, W. Zhao, X. Yin, C. Hou, C. Zhang, W. Ge, J. Zhang, Y. Wang, C. Zhou, and G. Yang, "The sunway TaihuLight supercomputer: System and applications," *Sci. China Inf. Sci.*, vol. 59, no. 7, pp. 1–16, Jul. 2016.

[24] J. Dongarra. (Jun. 2016). *Report on the Sunway Taihulight System*. [Online]. Available: https://www.netlib.org