Optimizing Vertical Link Placement and Congestion Aware Dynamic Elevator Assignment for Partially Connected 3D-NoCs

Yuxiang Fu, *Member, IEEE*, Chuan Zhang, *Member, IEEE*, Wenqing Song, *Student Member, IEEE*, Qinyu Chen, *Student Member, IEEE*, Hui Chen, *Graduate Student Member, IEEE*, Minghao Zhou, and Li Li, *Member, IEEE*

***Abstract*—The fully connected 3D-NoCs in which all routers are vertically connected with their neighbors above and below need a lot of Through-Silicon-Vias (TSVs), and they will occupy a large silicon area and reduce the fabrication yield. Thus, the idea of partially connected 3D-NoCs has emerged. The optimal number and placement of the vertical links (elevators) must be determined at the chip design stage, which is a multiobjective optimization problem of the performance and the cost. However, optimizing the static elevator placement needs a great amount of calculation and we can not examine all possible solutions at design time. Therefore, we propose a hybrid heuristic strategy for the static elevator placement and assignment, in which the genetic algorithm and the tabu search are combined. The dynamic assignment method is essential for the partially connected 3D-NoCs, and it leads to different traffic distributions and therefore has a huge impact on performance. Many previous static assignment methods can not dynamically change the elevator assignment according to the real-time states of the network, thus it may lead to network congestion. A congestion-aware dynamic assignment (CDA) scheme is proposed in this article, which considers the impact of the distance factor and the congestion factor on the network performance. Experiments show that the proposed CDA method can improve the network performance by 67%–86% compared with the random selection algorithm and can improve the reliability of the partially connected 3D-NoC as well. The key component for the CDA method, the path selection module (PSM), is implemented in FPGA, and the results show that its area cost is negligible compared with a router.**

***Index Terms*—3-D NoC, assignment, partially connected, placement, through-silicon-via (TSV), vertical links.**

I. Introduction

WHEN silicon scaling is approaching physical limits, it is not cost-economic anymore. Thus, apart from continuing to follow the more expensive technology, the alternative is to move in a completely different direction: 3-D integration. In addition, with the increasing number of components on a chip, the on-chip communications among components will dominate the performance, so the scalable communication backbone: Network-on-Chip (NoC) is the key. By introducing the vertical dimension, 3-D NoCs can shorten the communication distance and reduce the network energy due to the shorter average connection length compared with 2-D NoCs [1]–[3].

Most previous works were based on the fully connected 3-D NoCs [4], in which all the routers are vertically connected with their corresponding ones on the neighbor layers. However, a large number of Through-Silicon-Vias (TSVs) are required for the fully connected 3-D NoCs [5], which leads to an increase in the occupied silicon area and a decrease in the fabrication yield [6]. Moreover, the area consumed by the TSVs is orders of magnitude larger than the transistors that surround them. Therefore, to reduce the cost caused by the TSVs and to save more valuable silicon area for circuitry, we must limit the number of TSVs [7].

In this article, we will focus on the partially connected 3-D NoCs, in which only a part of routers are connected with their corresponding routers on the neighbor layers [8]–[13]. For the partially connected 3-D NoCs, it is necessary to determine the number and the locations of vertical links (elevators), which is a tradeoff between the performance and the cost. The placement problem must be solved at design time, which means the TSV placement can not be changed after the chip has been fabricated. At the design stage, it is impossible to make the design suitable for all the applications that may run on the chip. Thus, it is a practical way to choose some critical applications or "important" applications and apply them to a system simulator, then we collect the traffic statistics and use them to determine the TSV placement. However, optimizing the static elevator placement needs a great amount of calculation. Take a small 3-D NoC with a size of $4 \times 4 \times 4$ as an example, and we can evaluate the room for optimization as follows: we assume that there are four elevators in the 3-D NoC, then there are $C_{16}^{4} = 1820$ potential placements. And for each placement, there are 4^{64} different assignment, thus there are 1820×4^{64} possible solutions in the $4 \times 4 \times 4$ 3-D NoC, which is a big design space and we can not examine all possible solutions at design time. The

Manuscript received December 1, 2019; revised July 4, 2020 and September 27, 2020; accepted November 8, 2020. Date of publication November 16, 2020; date of current version September 20, 2021. This work was supported in part by the National Nature Science Foundation of China under Grant 61176024, and in part by Nanjing University Technology Innovation Fund. This article was recommended by Associate Editor A. K. Coskun. *(Corresponding author: Yuxiang Fu.)*

Yuxiang Fu, Wenqing Song, Qinyu Chen, Hui Chen, Minghao Zhou, and Li Li are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: lili@nju.edu.cn; yuxiangfu@nju.edu.cn).

Chuan Zhang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: chzhang@seu.edu.cn).

Digital Object Identifier 10.1109/TCAD.2020.3038338

design space will get bigger when the dimension of the 3-D NoC increases. Therefore, heuristic optimization methods are necessary.

Even though the TSV placement can not be changed at run-time, we can rely on adaptive assignment to adapt to applications that may not be considered at the design stage. Of course, we can compute the elevator assignment off-line for all known applications, but it does not work for those applications that we do not know at the design stage. Therefore, the dynamic elevator assignment is essential for the partially connected 3-D NoCs. As a limited interlayer communication medium, the elevator is prone to congestion and is critical to network performance. Besides, the unbalanced traffic distribution is another major cause of network congestion, which may result from an unreasonable elevator placement or assignment. However, many previous works regarding the elevator assignment methods just consider the distance factor and ignore the congestion information [11], [14]–[16].

To summarize, in this article, we focus on the study of the static elevator placement and the dynamic assignment, which will be significant to the practical application of partially connected 3-D NoCs.

The main contributions of this article are as follows.

1) We propose a hybrid strategy of the genetic algorithm (GA) and the Tabu Search (TS) for the static elevator placement and assignment. We design a method to encode an elevator placement solution in GA and a method to encode an assignment solution and its neighbor solution in TS. Besides, we provide a new fitness function to perform the optimization of the static elevator assignment and a lightweight analytical performance model for the design-time optimization.
2) We propose a congestion-aware dynamic elevator assignment scheme. We take both the distance and the congestion status into account when we solve the dynamic assignment problem. Thus, the network delay and the sum of the square of the buffer utilization from the source node to the assigned elevator is used as the assignment criterion. For each packet, we find the path of the minimum assignment criterion at the source node, and we assign the elevator in the path to the packet. Based on the analysis of the partially connected 3-D NoCs, we can reduce the hardware cost of the path selection module (PSM), and we show the architecture of the PSM and provide its implementation cost in FPGA.

The remainder of this article is organized as follows. Section II briefly reviews related works; Section III describes the static vertical link placement optimization based on the hybrid strategy of the GA and the TS; Section IV details the congestion-aware dynamic elevator assignment algorithm; Section V presents the simulation methodology and experiment results; Finally, Section VI concludes this article.

II. Related Works

In the partially connected 3-D NoCs, only a part of routers in the adjacent layers are connected. The optimal number and placement must be determined at the chip design stage. Xu *et al.* [17] thought that a good elevator placement needs to satisfy certain requirements: 1) each node has at least one elevator within the distance range $h_{\max}$; 2) the number of elevators should be as small as possible; 3) the elevators should be uniformly distributed; and 4) the redundant elevators are required for fault-tolerant and there should be as many elevators as possible in the distance range, and $r_{\min}$ and $r_{\max}$ are the minimum and maximum number of elevators in the distance range of each node. The first step is to determine the minimum number of elevators for the given parameters (topology, $r_{\min}$ and $h_{\max}$), and the second step is to optimize the maximum redundancy according to the three parameters given above and the number determined in the previous step. This method considers the cost and reliability-oriented optimization, but does not take the optimization of network performance into account. In addition, the method needs to determine $r_{\min}$ and $h_{\max}$ in advance. These two parameters have a great impact on performance, but the author has not provided a method to obtain these two parameters. Moreover, in the process of the location optimization, the default elevator assigned to each node is the nearest one, which is simple but does not take into account the actual load of the network.

The goal of the works [18], [19] is to minimize the communication cost by a suitable TSV placement and an optimized mapping of cores to routers. Therefore, the TSV placement problem is solved together with the mapping problem by using the TLP or the PSO algorithm. However, these two works did not take the TSV cost as an optimization objective, and they both restricted the vertical connectivity to only 25%. In our method, the network performance and the TSV cost are optimized simultaneously. Besides, in the optimization process of the TSV placement and mapping in [18] and [19], the assignment of the routers to the vertical links is just determined based on the shortest distance and ignores the network status. In our method, the distance and the network congestion information are both considered to solve the assignment problem.

Once the placement of the elevators is determined, how to assign elevators dynamically according to the varying traffic loads is very important to performance. Therefore, many previous works focus on it. In [20], the elevator assignment to a packet is random, which brings in uniform traffic distribution. However, the random assignment only works when the LEAD routing algorithm is applied, because packets are free to choose any elevator to transmit in the LEAD algorithm. Nevertheless, it needs extra virtual channels along the X and Y dimensions, which will result in more area and power overhead.

The TS algorithm is used in [14] for the elevator assignment optimization. However, it has high computational complexity. Especially for large-scale problems, it may take several hours. Thus, it cannot change the elevator assignment in real-time. It means that before a new application, a reconfiguration step is needed to run the TS to optimize the elevator assignment. This approach is not practical, because once the application changes, the time consumed to reconfigure the assignment is very long. Therefore, in this article, we only employ the TS for the static elevator assignment.

In [16], the elevator assignment is based on the distance between the elevator and the current router. For each router, its nearest elevator's location information needs to be stored, and it can be represented in 12 reconfigurable bits. These reconfigurable bits guide the packets to find the nearest elevator in the right direction.

A real-time distance-based elevator assignment approach is employed in [15]. The elevator selection process can be divided into four steps. In the first step, the elevators in the routing paths with the shortest overall Manhattan distance are selected. And the overall Manhattan distance is the distance from the source node to the destination node. If the number of these selected elevators is greater than one, it will go the second step. In the second step, the elevators with the shortest Manhattan distance to the source node will be selected and if there is more than one elevator, we will go to the third step, in which the elevators with the shortest distance along the X dimension to the source node will be selected. If all of the above steps can not identify a unique elevator, the elevator that is not in the same Y-half-plane as the source node will be selected. The four-step process narrows down the elevator choices but it is time consuming and ignores the congestion status.

In [7], one elevator is selected randomly between the nearest routers in which the corresponding vertical link is not removed. Therefore, the elevator assignment method in [7] also chooses the distance between the elevator and the current router as the selection criterion. The Load-Balancing of Up-Down Routing presented in [21] chooses different root nodes in different layers to distribute the potential hotspots near the roots, which is a very easy way to balance the traffic distribution among the critical medium like TSVs in 3-D NoCs and is similar to the random selection of elevators. Moreover, the layered shortest path routing (LASH) proposed in [21] intends to find the true shortest-path routing in irregular topologies, thus if we apply it on the partially connected 3-D NoC, the elevator in the path with the shortest overall distance will be selected.

The elevator assignment approaches proposed in [7], [15], [16], and [21] are based on the Manhattan distance and ignore the congestion information. To address this problem, we propose a congestion-aware dynamic elevator assignment method, in which both the distance factor and the congestion factor are considered.

III. Optimizing Vertical Link Placement Based on Hybrid Strategy of the Genetic Algorithm and the Tabu Search

The number and placement of the elevators are closely related to the performance and the cost of the 3-D NoC system. The goal of the optimization is to improve the performance and to reduce the cost of the partially connected 3-D NoC systems, and it is a multiobjective optimization problem. A reasonable solution to a multiobjective problem is to investigate a set of solutions called Pareto optimal set, each of which cannot be improved with respect to any objective without worsening at least one other objective. Classical optimization methods only find one solution in one simulation run, thereby they are inconvenient to explore a large design space. The GA, a population-based approach, can find multiple solutions in one single simulation run. In addition, most multiobjective GAs do not require the user to prioritize, scale, or weigh objectives. Therefore, the GA is an ideal candidate for solving multiobjective optimization problems.

During the placement optimization, for a given number/locations of elevators, the performance of the partially connected 3-D NoC can be considerably affected by the assignment of elevators. Therefore, in this article, we employ the TS to solve the static assignment optimization problem, which is a commonly used approach for assignment problems [14]. Thus, the hybrid strategy of the GA and the TS [22] is adopted in this work to find Pareto solutions for the designers to enhance design flexibility.

The hybrid strategy incorporates the GA and the TS, and the rationale behind using such a hybrid method is to combine the diversified global search and the intensified local search capabilities of the GA and the TS, respectively. The GA is employed to optimize the elevator number and the placement. For a given number/locations of elevators, the static elevator assignment is optimized by the local search method: TS.

The GA is an iterative process, which usually starts from a population P_0 of N_{pop} randomly generated individuals. Each iteration consists of evaluation, selection, crossover, mutation steps, and the population in each iteration is called a generation. In the evaluation step, the fitness (detailed in Section III-C) of every individual in the population is evaluated using the TS method. In the selection step, individual genomes with high fitness are chosen from a population for later breeding. Then, the second-generation population can be generated from those selected genomes through a combination of crossover and mutation. The newly generated candidate solution will be used in the next iteration. And the iterations stop after reaching the maximum generation or obtaining a satisfactory fitness.

The process of the hybrid strategy applied to optimize the static elevator placement can be illustrated by the flowchart diagram in Fig. 1.

A. Encoding Placement Solution in Genetic Algorithm

In fact, GAs work on the coding of a problem rather than the problem itself. The proposed chromosome representation encodes an elevator placement solution (number/locations) as an ordered sequence of binary digits, where each gene stands for one alternative elevator location. The value of the gene i is "1" which means an elevator exists on the router i, otherwise, no elevator exists on that location. For example, the sequence "0001001011000000" means that elevators exist on routers 3, 6, 8, 9.

B. Encoding Assignment Solution and Its Neighbor Solution in Tabu Search

In the GA, each chromosome represents a case of the placement of elevators. The fitness value of each chromosome is required before moving on to the selection stage, and it relies on the assignment of elevators, which can be obtained

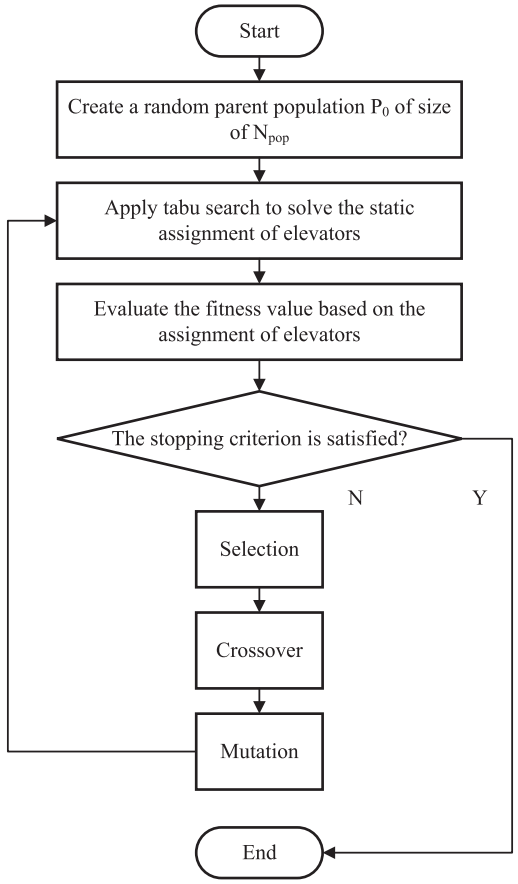


Fig. 1. Process of the hybrid strategy applied to optimize static elevator placement.

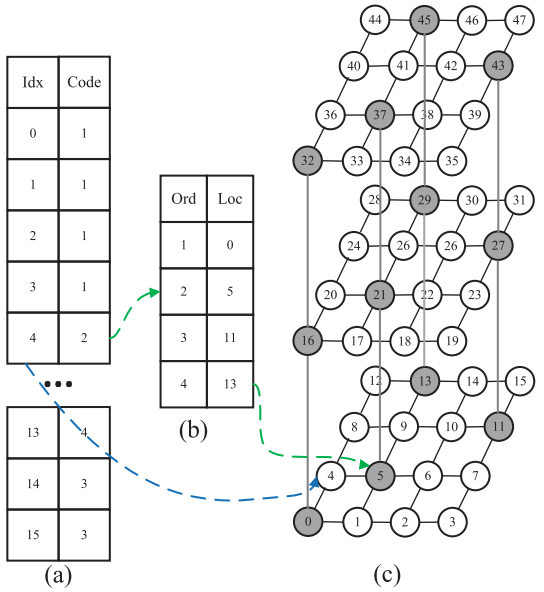


Fig. 2. Example of the decoding of an elevator assignment solution, (a) encoding of an elevator assignment solution, (b) look-up-table, and (c) given number/locations of elevators for 3D-NoCs.

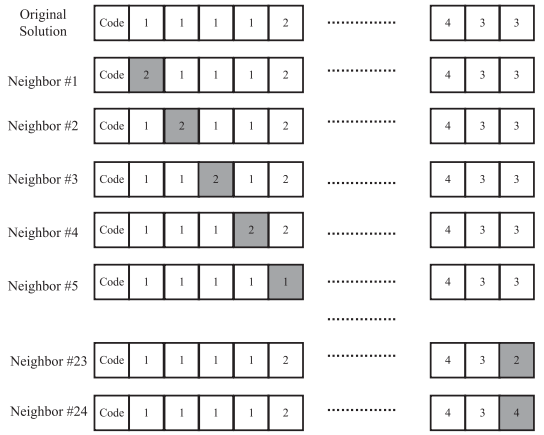


Fig. 3. Example of the closest neighbors of a solution.

using the local search method: TS. In the TS, the solutions are also encoded as an ordered sequence of integer numbers from 1 to M, where M is the number of elevators. The integer denotes the ordinal of an elevator, and a look-up-table recording (*ordinal*, *location*) pairs are required to find the elevator location according to an ordinal value. For example, as depicted in Fig. 2, in a solution, the code corresponding to router 4 is 2, and in the look-up-table, the number paired to 2 is 5, which means that the router 4 is assigned to the elevator on router 5.

In this article, we define the closest neighbor of a solution as the one in which only one integer is different from the solution and the difference is 1 or -1. Therefore, for a solution with the code length of L, the number of neighbors is up to $2L$, and an example is shown in Fig. 3. The neighbor solution is required in TS for the next search, and the neighbor solution presented here means that it only has one different router assignment compared with the current solution, and the change in the assigned elevator's ID is $+1$ or -1. We invalidate the solution if the assigned elevator's ID is less than 1 or larger than the number of the elevators.

In the figure, for "neighbor #1," only the first integer is changed and the rest integers are the same as the original solution. Specifically, the first integer is changed from 1 to 2 in neighbor #1. Similarly, "neighbor #2" change the second integer, "neighbor #3" changes the third integer and so on.

C. Fitness Function and Analytical Performance Model

To perform the optimization of the static elevator assignment to achieve the best fitness value, the fitness function Obj_{pef} can be set as the weighted sum of the average hops L_{AvgHops} and the traffic load variance T_{var} as follows:

$$\min \{Obj_{\text{pef}} = aL_{\text{AvgHops}} + bT_{\text{var}}\} \tag{1}$$

where a and b are weights, and $a+b=1$. For example, if we take the average network delay T_{AvgDelay} as the optimization objective, then before the optimization process, the weights a and b can be obtained through regression methods for the three variables T_{AvgDelay}, L_{AvgHops}, T_{var}, which can be collected from simulations.

L_{AvgHops} refers to the average hops, which are determined by the distances of all the source–destination pairs and their

communication frequency

$$L_{\text{AvgHops}} = \sum_{i=0}^{N-1} \sum_{j=0, j\neq i}^{N-1} f_{ij} \times Dis(i,j)$$
$$\text{where } f_{ij} \in (0,1), \sum_{i=0}^{N-1} \sum_{j=0, j\neq i}^{N-1} f_{ij} = 1. \quad (2)$$

N is the total number of the nodes in a 3-D NoC, and f_{ij} is the communication frequency ratio between the source node i and the destination node j, and $\text{Dis}(i,j)$ denotes the distance between the nodes i and j. N and $\text{Dis}(i,j)$ are related to the dimension of the 3-D NoC, and f_{ij} depends on the applications applied to the 3-D NoC system.

T_{var} refers to the variance of the traffic loads, which represents the traffic distribution uniformity

$$T_{\text{var}} = \sum_{i=0}^{N-1} \frac{(T_i - T_{\text{avg}})^2}{N} \quad (3)$$

where T_i and T_{avg} are the traffic load of router i and the average traffic load, respectively.

The first part of the fitness function L_{AvgHops} represents the ideal performance for the network without congestion; and the second part indicates the network's congestion level. In short, the congestion occurs at a router that many paths will route through. And the path distribution determines the traffic loads of routers. Therefore, if there are two same networks with the same packets injection rate (PIR), which means that they have the same total traffic loads in a fixed time, then the more unbalanced traffic loads are, the more congested the network will be. In other words, if the network has a poor distribution uniformity (i.e., large T_{var}), the level of congestion will grow, which leads to the degradation of the network saturation threshold.

The average hops L_{AvgHops} and the variance of the traffic load T_{var} are obtained through an analytical model as shown in Algorithm 1. Of course, L_{AvgHops} and T_{var} can also be obtained through simulators, but there will be a huge number of searches in the optimization process, and the optimization process will become extremely long if we invoke a simulation to get L_{AvgHops} and T_{var} at each search. Utilizing the analytical model is a common method in the optimization of SoC designs, which can effectively shorten the time of the optimization process.

In Algorithm 1, the intralayer routing algorithm in this article is *XY*, and the deadlock-free rule for the elevator selection uses the method proposed in [23]. Besides, the traffic distribution is represented by $f(i,j)$. There are two ways to obtain $f(i,j)$: the first way is to assume a synthetic traffic distribution, such as $f(i,j)$ is equal for any i and j in the uniform pattern; the second way is to perform many applications on the system simulators, such as Gem5 [24], and then collect statistical data from these simulators to form an $f(i,j)$ table whose elements are averaged among different applications.

Algorithm 1 Analytical Performance Model

Input: 3D NoC size $N = X \times Y \times Z$, $f(i,j)$ is the communication frequency ratio between the source node i and the destination node j, T_{total} is the total packets injected into the network

Output: $Obj_{\text{pef}} = aL_{\text{AvgHops}} + bT_{\text{var}}$

```
sum_L = 0
Traffic[N] = 0
for i = 0; i < N; i++ do
    for j = 0; j < N; j++ do
        if j ≠ i then
            sum_L += D(i,j) × f(i,j),
            // D(i,j) is the Distance between node i and node j through the assigned elevator;
            for all node k in the path between node i and node j do
                Traffic[k] += f(i,j) × T_total
            end for
        end if
    end for
end for
L_AvgHops = sum_L / ((N)×(N−1))
T_var = std.Dev(Traffic)
Obj_pef = aL_AvgHops + bT_var
```

IV. Congestion Aware Dynamic Elevator Assignment

For 3-D NoC systems, elevators are transport channels across different layers. In this article, selecting a specific elevator to go upward/downward for a packet is called an elevator assignment. The vertical link placement [17] is determined before the chip fabrication, thus it is static. Dynamic elevator assignment is essential technologies to adapt to different applications at run-time. The elevator utilization imbalance, which is caused by the unbalanced elevator distribution or unbalanced assignment, presents different characteristics for different applications. Therefore, how to find a dynamic assignment method that can balance elevator utilization for various applications is a great challenge.

A. Adopted Routing Algorithm

Essentially, the elevator assignment is a part of the routing algorithms specific for partially connected 3-D NoCs, which means that different elevator assignment will lead to different routing paths. For routing algorithms, it is very important to avoid deadlock. Some deadlock-free rules for elevator selection are proposed in [23], which eliminate the need for virtual channels, thus greatly reducing area and power overhead. This article adopts ruleset B proposed in [23]: (1) if the source node and the destination node are at different layers, any elevator in the south-or-due-east of the source node (including itself) can be selected; (2) if any elevator cannot be found according to Rule (1), we will choose a pivot elevator that has no other elevators in its south-or-due-east. The deadlock-free proof of the above rules can be referred to [23]. Fig. 4 shows an example of the valid region for the elevator

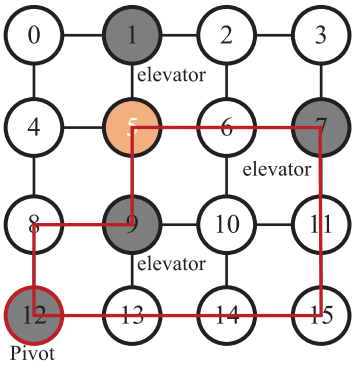


Fig. 4. Valid region of the elevators and the pivot elevator.

selection and the pivot elevator. Any elevator in the region within the red line can be selected for node 5. The elevator located on the node 12 is the pivot elevator because it is the southernmost elevator and there is no other elevator to its east.

The assigned elevator acts as an intermediate node, and we employ XY routing between the source node and the intermediate node (on the source layer) or between the intermediate node (on the destination layer) and the destination node.

B. Dynamic Elevator Selection

In the case of more than one elevator in the valid region, how to select one of them as the vertical channel to improve performance is a major problem to be solved in this article. In previous works, some criteria for the elevator selection have been proposed, such as the simple random selection or the Manhattan-distance-based selection. As we all know, the delay of data transmission is not only related to the Manhattan distance between the source node and the destination node, but also related to the network congestion along the transmission path. Once the traffic allocated to the elevator exceeds the maximum capacity it can transmit, congestion will occur in the elevator, and packets will have to wait a long time to pass through. Congestion will spread in the network, resulting in a great degradation in network performance. The Manhattan-distance-based selection method ignores the network congestion, and the random selection method does not even consider the distance factor, so both methods have shortcomings. To this end, we propose a congestion-aware dynamic elevator assignment method. In this method, we use the weighted sum of the router delay and the square of the input buffer utilization of each router along the path from the source node to the assigned elevator as the selection criterion as follows:

$$vl_{\text{sel}} = \arg\ \min_{vl \in VL}\left(\sum_{l \in L_{vl}} (\alpha d + \beta Buf^2)\right) \tag{4}$$

where vl_{sel} is the elevator that we selected, VL is the set of the eligible elevators, L_{vl} is the path from the source node to the destination end of the assigned elevator vl, d is the router delay, Buf is the utilization of buffers on the routing path, α and β are weights and $\alpha + \beta = 1$. The weights α and β in (4) are trained through a number of training simulations, in which the weights can be adjusted to reduce the average network latency.

The average network delay, which is one of the most important metrics of the network performance, consists of three major parts [shown in (5)]: 1) the delay consumed on the path from the source node to the elevator t_{s2v}; 2) the delay of the elevator t_v; and 3) the delay spent on the path from the elevator to the destination node t_{v2d}

$$t_{\text{avg\_net\_}lat} = t_{s2v} + t_v + t_{v2d}. \tag{5}$$

Because the adjacent layers are partially connected, which means that only a part of routers have their local elevators, the traffic between the source nodes and the elevators shows a many-to-few scenario, and many routers will compete for few elevators. Conversely, the traffic from the elevators to the destination nodes presents a few-to-many pattern. Therefore, if the traffic is evenly distributed between layers, the delay consumed on the path from the source node to the elevator contributes mostly to the network delay, and reducing it is the key to performance improvement.

If one destination layer has more traffic than other layers, then the routing path in the destination layer will contribute most to the network delay. However, the congestion in the destination layer will spread back to the elevator, and eventually back to the source layer. Thus, in this case, the congestion status of the elevators and the path in the source layer can also reflect the network delay. Because the elevator needs to be selected when a packet is generated at the source node, it is much easier and cost-effective to just consider the path in the source layer. The buffer information of the path in the destination layer has to be transmitted through the elevator, which will increase the elevator congestion. Therefore, to reduce computational complexity and communication traffic, we just consider the path from the source node to the elevator. In that case, only the congestion information of the nodes on the same layer needs to be stored, communicated, and updated. Besides, the congestion information of elevators from the source layer to the destination layer is also considered, because the elevators are limited resources and are easily prone to congestion.

In addition, when we calculate the selection criteria, we use the square of the buffer utilization instead of the direct buffer utilization, in order to better emphasize that the possibility of congestion caused by adding new data into a high-utilization buffer is greater than one caused by adding new data into a low-utilization buffer. It should be noted that only the utilization of the input buffer of the router that the packet will pass through can be used to calculate the congestion criterion. Fig. 5 shows a simple example. Packets are sent from the node 4 to the node 20, and the congestion criterion is calculated based on the path from the node 4 to the node 25. Therefore, the buffers that the packets pass through are the west input port of the router 5, the north input port of the router 9, and the down input port of the router 25.

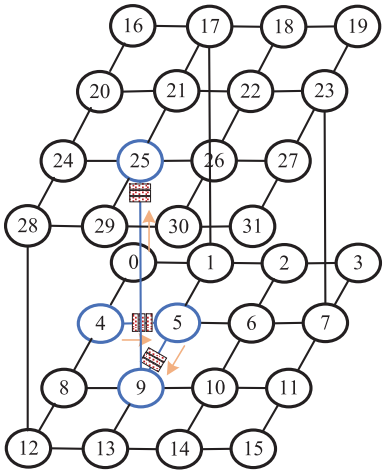


Fig. 5. Buffers used in the congestion criterion.

C. Process of Congestion-Aware Dynamic Elevator Assignment and Implementation Analysis

In this article, the dynamic elevator assignment is based on the source and the destination of a packet and the congestion information of the network, which is independent of the current node location, so different packets have different elevator assignment. The process of the congestion aware dynamic elevator assignment can be illustrated by the flowchart diagram in Fig. 6. When a new packet is generated at the source node, the ID information of the elevator to be passed through is calculated and added into the packet header. The elevator assignment process mainly includes comparing the congestion criteria of all the candidate paths that packets may go through and then selecting the path with the minimum congestion criterion.

The PSM consists of two multipliers, one adder, one accumulation unit, one comparator, and one table storing buffer utilization as shown in Fig. 7. The number of the accumulation, which equals to the length of the path from the source to the elevator $O(m+n)$, is restricted by the dimension of the layer. Besides, due to the valid region determined by the deadlock-free rule adopted in this article, the number of comparing, which equals to the number of eligible elevators in the valid region $O(\nu \times m \times n)$, where ν is the fixed elevator ratio, is limited. Therefore, the calculation time of path selection is $O((m+n) \times (\nu \times m \times n))$.

For each router, the input buffer's utilization of other routers on the same layer needs to be stored. It should be noted that the buffer utilization is not updated every cycle, but averaged over an interval. The interval is a tradeoff between the computation frequency and the granularity of the congestion control. And the interval can not be too short to complete the path selection and can not be too long to alleviate congestion in time. Thus, the interval time must be longer than the processing time of the path selection, which includes the multicast time of the traffic information $O(m+n)$ and the calculation time of the path selection $O((m+n) \times (\nu \times m \times n))$, so it can be set to be $O((m+n) \times (\nu \times m \times n))$. At the beginning of each interval, the average buffer utilization of each router's port is calculated,

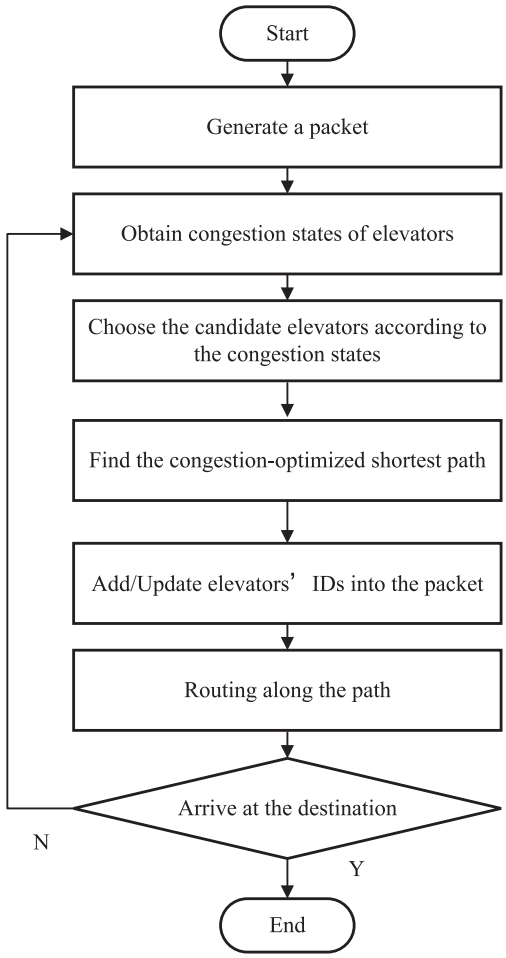


Fig. 6. Process of the congestion aware dynamic elevator assignment.

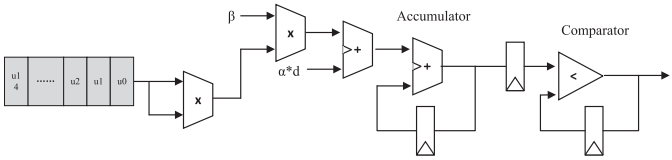


Fig. 7. Architecture of the PSM.

and then it is multicast to other routers on the same layer for updating.

As Fig. 6 shown, the elevators must be selected before the newly generated packets being injected into the network. In order to avoid new packets waiting in the injection queue for a long time, we build an elevator selection look-up table to store the elevator for each destination. Thus, when a new packet arrives, the elevator will be selected according to the table, and at the same time, the destination of the packet will be pushed into a FIFO if the PSM is busy, and the table will be updated after the calculation has finished. Since then (must in the same interval, which means the buffer information is the same), another packet with the same destination will use the updated table, and the PSM can be bypassed. Therefore, it should be noted that the calculation time of the PSM will not affect the execution time of an application.

Due to the loose requirement for the computation latency, the implementation area can be further reduced at the cost of

TABLE I
PSM IMPLEMENTATION

Resource	Router	PSM	Ratio of PSM to Router
LUT	11828	152	1.3%
FF	2656	91	3.4%

TABLE II
CONFIGURATIONS OF THE HYBRID STRATEGY OF THE GA AND THE TS

Configurations	Value
population size	48
generations	200
Number of objective functions	2
Number of bits for binary genes	16
Probability of crossover	0.6
Probability of mutation	0.01
Seed for random number generator	0.0345
Length of the tabu table	$6 \rightarrow 13$ random
The solution size in tabu search	16
The number of iterations in tabu search	$10 \times N$ (N is the number of elevators)

increased delay. Moreover, because we employ the XY routing algorithm for the intralayer routing, for each router, only one input buffer utilization of each other router on the same layer needs to be stored. Therefore, there are only $m \times n - 1$ entries recording other routers' buffer utilization for each router in the $m \times n \times k$ 3-D NoC. To evaluate the area cost of the PSM, we implement it and a 5 port router on Spartan6-xc6slx150t, and provide their resource utilization in Table I. From the table, we find that the area cost of the PSM is negligible compared with a router, which is consistent with the analysis above.

Actually, each node needs to send one packet containing its buffer information to other $m \times n - 1$ nodes. Thus, the equivalent additional PIR that results from buffer information multicast is $(m \times n - 1/T_{\text{interval}}) = (m \times n - 1/(m+n) \times (\nu \times m \times n)) \approx (1/(m+n) \times \nu)$ packet/node/cycle, which means when the network dimension increases, the traffic overhead and the interval time both increase, but the interval time has more increase. Therefore, the additional PIR that results from buffer information multicast will decrease with the increasing network size and it will not lead to the scalability problem.

V. EXPERIMENTAL RESULTS

In this section, we first describe the performance evaluations of the hybrid elevator placement method, and then we present the evaluations of the congestion-aware dynamic elevator assignment.

A. Evaluations of Elevator Placement Optimization Based on the Hybrid Strategy of the Genetic Algorithm and the Tabu Search

In the experiments, we apply the hybrid strategy of the GA and the TS to a partially connected 3-D NoC with a size of $4 \times 4 \times 4$. The configurations of the hybrid strategy of the GA and the TS are shown in Table II.

The hybrid algorithm ran on a computer with Linux OS, Intel Core i7-3770 CPU @ 3.4 GHz, and 8-GB RAM,

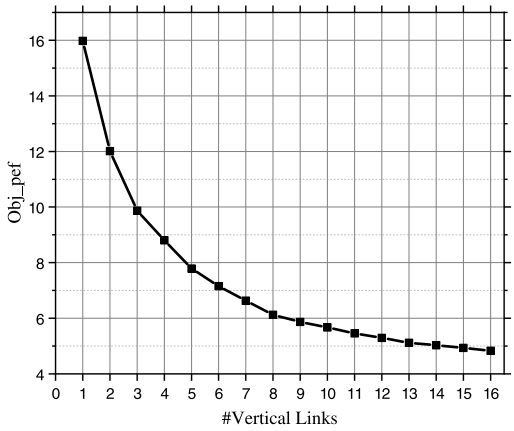


Fig. 8. Pareto frontier of the number of elevators and network performance.

whose convergence time is 400 min. For a fair comparison with the well-studied multiobjective optimization algorithm, AMOSA [25], we defined 32 variables, in which the first 16 variables are boolean that represent whether an elevator exists in that node, and other variables are integers that indict the indexes of the elevators assigned to each node. And the 16 boolean variables are the same as the placement solution definition used in the GA algorithm in Section III-A and the 16 integer variables are the same as the assignment solution definition used in the TS algorithm in Section III-B. Besides, the objective functions used in the AMOSA algorithm are also the same as the fitness function proposed in Section III-C. The elevator placement optimization can be seen as the number/location optimization problem nested with the assignment optimization problem. Thus, in this article, we use the hybrid algorithm in which the GA algorithm is nested with the TS algorithm. And the GA algorithm is used for optimizing elevator locations, meanwhile, the TS algorithm is responsible for the assignment when elevator locations are given. For any location solution searched by the GA algorithm, its optimal assignment can be obtained by the TS algorithm. But, in the AMOSA algorithm, the elevator locations and their assignment are updated at the same time, and some solutions with a larger number of elevators may be removed from the archive because it has no optimal assignment. It will lead to repeat searches and it is hard to cover the number range of elevators. When running in the same computer with the proposed hybrid algorithm, the AMOSA algorithm's convergence time is 412 min, which is slightly longer than that of the proposed algorithm. Besides, it just provides 12 nondominated solutions, but four solutions corresponding to four different elevator numbers are missed.

Fig. 8 is the Pareto frontier of the elevator number and the network performance obtained by the hybrid strategy. Each point in the Pareto frontier indicates the optimal network performance for the corresponding elevator number. In the figure, x-axis indicates the number of elevators, whose range is $1 - 16$, and y-axis represents the weighted sum of the average latency and the variance of the traffic load as shown in (1).

As shown in Fig. 8, with the increase of the elevator number from 1 to 16, the value of the objective function decreases continuously, which means that network performance improves

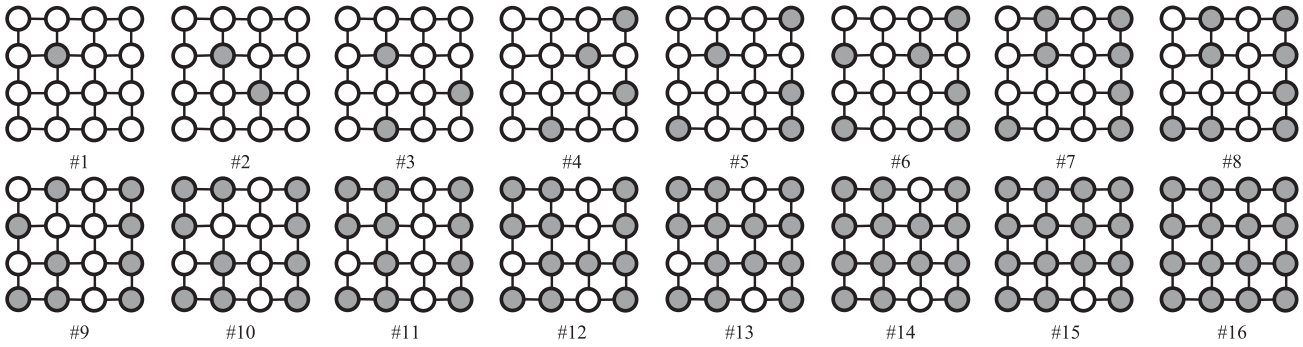
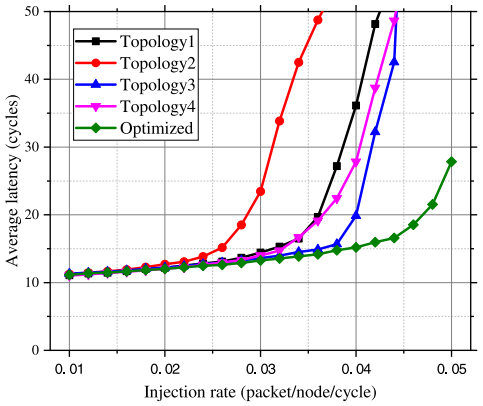
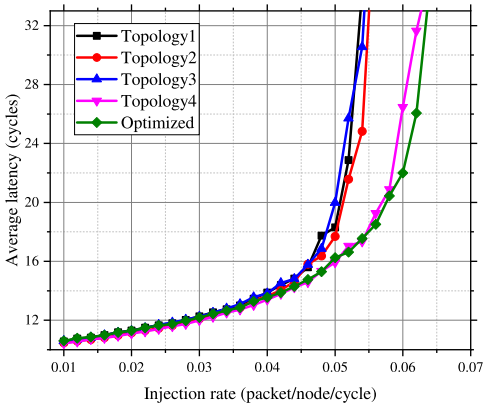


Fig. 9. Elevator placements with different numbers.



(a)



(b)

Fig. 10. Average network latency of different topologies. (a) 8 elevators. (b) 12 elevators.

continuously. On the one hand, with the increase of the elevator number, the vertical bandwidth between layers is also increasing; on the other hand, the ratio of the vertical channel number to the router number is increasing, which can effectively reduce the competition for vertical channels and makes the network traffic distribution to be more balanced. In addition, according to Fig. 8, we find that the gain of network performance decreases with the increasing number of vertical channels.

The optimization of the number and the placement of elevators is a multiobjective optimization problem, in which the cost and the NoC performance need to be optimized simultaneously. The cost of the chip is monotonously increasing with the number of elevators, thus in the x-axis of Fig. 8 the number of elevators is used to represent the cost of the chip, which means that Fig. 8 can be regarded as the Pareto frontier of the chip cost and the network performance. Therefore, Fig. 8 can provide design references for 3-D NoC designers.

TABLE III
CONFIGURATIONS OF THE SIMULATOR

Configurations	value	Configurations	value
size	$4 \times 4 \times 4$	routing algorithm	Redelf [23]
frequency	1 GHz	virtual channels	w/o
buffer depth	5 filts	Warm-up time	5000 cycles
packet size	5 filts	temporal distribution	Poisson
flow control	wormhole	arbitration strategy	round-robin

Each point along the Pareto frontier in Fig. 8 corresponds to the optimal elevator placement for a corresponding elevator number. The optimal elevator placement examples with the number of 1 to 16 are shown in Fig. 9.

To evaluate the placement (essentially a topology) obtained by the hybrid search algorithm proposed in this article, we compared it with several randomly generated topologies. For simplicity, in this section, when implementing the hybrid algorithm of the GA and the TS algorithm, the following assumptions are adopted in the analytical model: the probability of communication between all nodes is equal. (The assumption is not mandatory. We just made such an example assumption in the analytical model to show the effectiveness of the hybrid strategy of the GA and the TS. Other assumptions can also be applied to the analytical model.) Therefore, in the simulation setup, we use the corresponding uniform traffic mode. In addition, a simple vertical channel allocation algorithm [16] is adopted in the simulation process: the selection is based on the distances from the current node to the elevators. The experiment results with eight elevators and 12 elevators are shown in Fig. 10. The average network latency of the optimized topology and other four randomly generated topologies are shown in the figure. We find that the optimized topology has a lower average network delay and higher saturation throughput than the randomly generated topologies. This advantage is more obvious when the number of elevators is small, because the optimization of the topology focuses on the distribution of network traffic loads, and when the number of elevators is small, the network is more prone to congestion. In

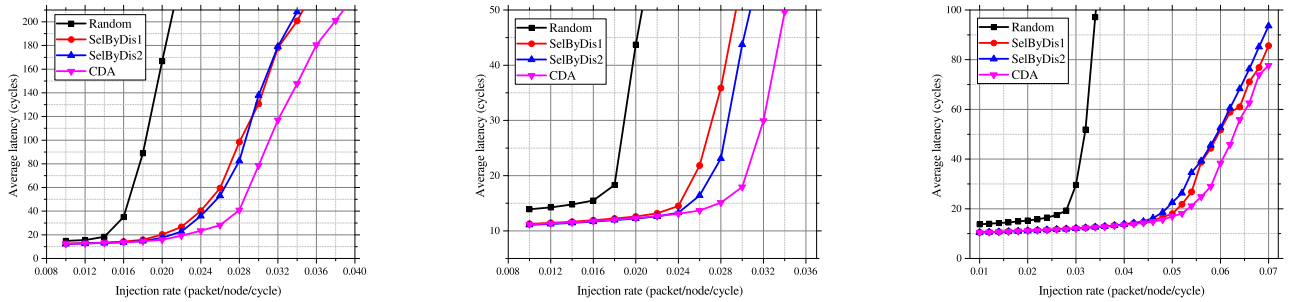


Fig. 11. Average network delay under uniform traffic pattern with different elevator densities. (a) 25%, uniform (b) 50%, uniform (c) 75%, uniform.

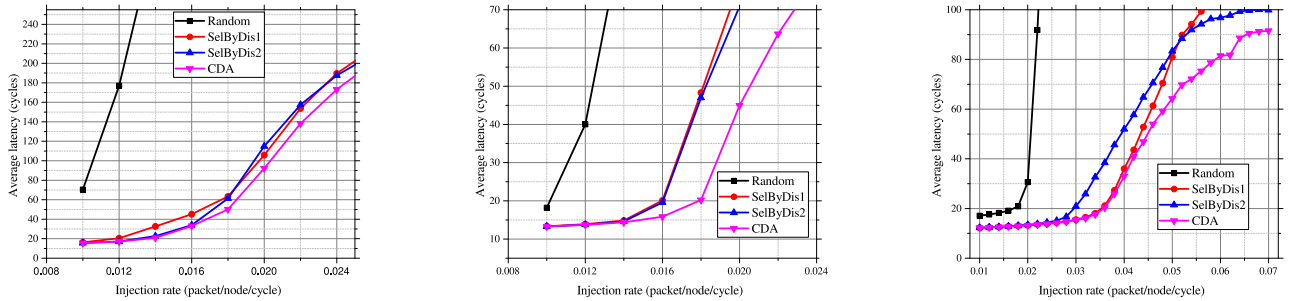


Fig. 12. Average network delay under transpose traffic pattern with different elevator densities. (a) 25%, transpose (b) 50%, transpose (c) 75%, transpose.

this case, the performance improvement caused by balancing traffic loads will be more obvious.

B. Evaluations of Dynamic Elevator Assignment Methods

The simulation is based on the 3-D NoC simulator AccessNoxim [26], which combines the network simulator Noxim [27], a power consumption model, and the thermal model HotSpot [28]. In addition, we have also made some modifications to the simulator, adding a congestion-aware elevator assignment controller and a new routing algorithm Redelf [23]. Moreover, the original simulator does not support the partially connected 3-D NoCs, so we construct a partially connected 3-D NoC by disabling some vertical links. The simulations in this article are based on a 3-D NoC with a size of $4 \times 4 \times 4$. The buffer depth and the packet size are both 5 flits. In this article, we run the simulations based on a variety of synthetic traffic patterns, such as transpose, uniform, shuffle, hotspot, bit-reversal, and several real application traffic. Table III shows the parameter configurations of the simulator.

In Section II, two previous related works about the elevator assignment methods have been presented: 1) the first elevator assignment method chooses the distance between the elevator and the current router as the selection criterion [16], and in this section it is referred to as SelByDis-1; 2) the second elevator assignment method is a process of repeated selections according to distance [15], and in this section it is referred to as SelByDis-2. To evaluate the performance of the proposed congestion-aware dynamic elevator assignment method, we compared it with the random selection, SelByDis-1 and SelByDis-2. For fairness, all four elevator assignment methods are based on the same routing algorithm Redelf [23]. Besides, to evaluate the performance of these elevator assignment methods under different application scenarios, we applied these four assignment methods to the partially connected 3-D NoCs under traffic patterns like uniform, transpose, shuffle, hotspot, and bit-reversal. Moreover, in order to evaluate the impact of the network loads on the performance, we set the packet injection rates from 0.01 to 0.05 packets/node/cycle with the step size of 0.002 packets/node/cycle. In addition, to evaluate the performance of these elevator assignment methods under different elevator densities (the ratio of the number of elevators to the number of routers) and different topologies, we simulated four random topologies at 25%, 50%, 75% elevator densities, respectively.

Figs. 11–15 show the average network delay for different traffic patterns with the elevator densities of 25%, 50%, 75%. In the simulation configurations, there are 16 routers per layer, so the elevator densities of 25%, 50%, and 75% correspond to 4, 8, and 12 elevators, respectively. It should be noted that in order to eliminate the impact of different topologies on the performance, each data point in these figures is the mean of average network delays under four random topologies. As can be seen from Figs. 11–15, the performance of our method is superior to that of other methods (random, SelByDis-1, SelByDis-2) under uniform, transpose, shuffle, and bit-reversal traffic patterns, except that the performance of the proposed method is close to ones of the SelByDis-1 and the SelByDis-2 under the hotspot pattern in Fig. 14.

By comparing Fig. 11–15, we find that the average network delay decreases with the increase of the elevator density. The

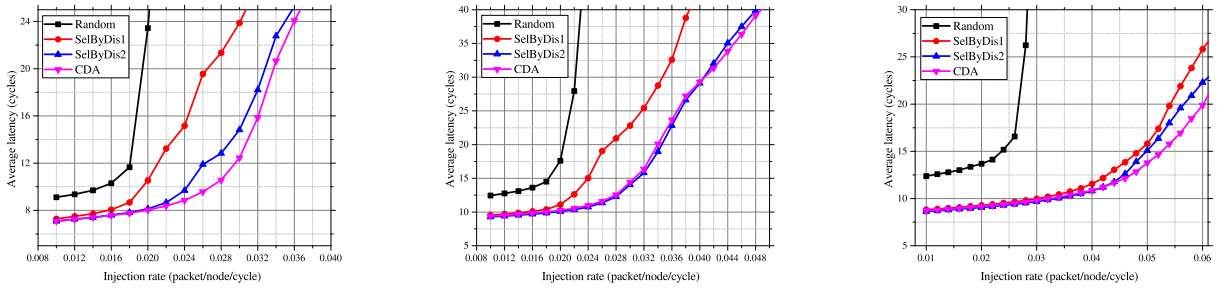


Fig. 13. Average network delay under shuffle traffic pattern with different elevator densities. (a) 25%, shuffle (b) 50%, shuffle (c) 75%, shuffle.

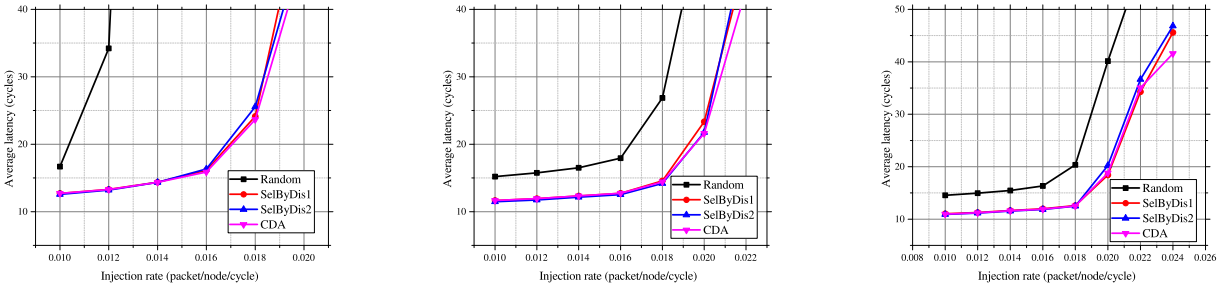


Fig. 14. Average network delay under hotspot traffic pattern with different elevator densities. (a) 25%, hotspot (b) 50%, hotspot (c) 75%, hotspot.

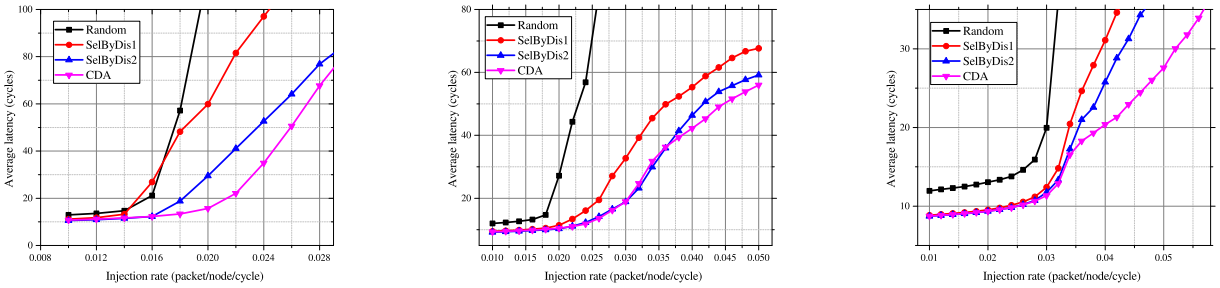


Fig. 15. Average network delay under bit-reversal traffic pattern with different elevator densities. (a) 25%, bit-reversal (b) 50%, bit-reversal (c) 75%, bit-reversal.

TABLE IV
COMPARISON OF DIFFERENT ASSIGNMENT METHODS

elevator density	random	SelByDis-1 [15]	SelByDis-2 [16]	CDA
25%	0.014 (1.86)	0.022 (1.18)	0.022 (1.18)	0.026
50%	0.018 (1.67)	0.024 (1.25)	0.026 (1.15)	0.030
75%	0.030 (1.73)	0.048 (1.08)	0.046 (1.13)	0.052
100%	0.070 $(XYZ Routing)$			

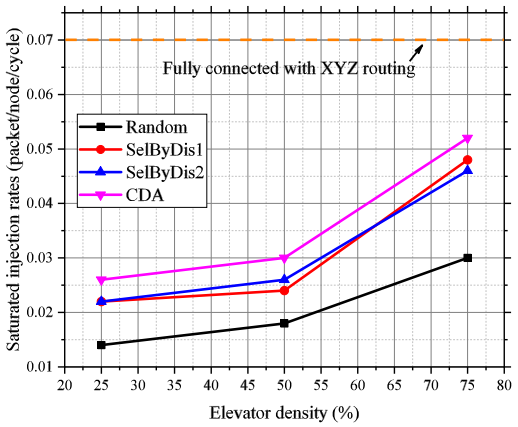


Fig. 16. Saturated injection rates of different assignment methods.

reasons are as follows: 1) with the increase of the number of elevators, the average distance from each node to the elevator decreases and 2) with the increase in the number of elevators, the competition for elevators will be alleviated.

Table IV presents a comparison of saturated injection rates for different elevator assignment methods. Here, we just use the uniform traffic pattern as an example, and other traffic patterns are similar. The saturated injection rate refers to the injection rate that causes the average network delay to suddenly increase, which means that after this injection rate, the communication capacity of the network reaches saturation. In

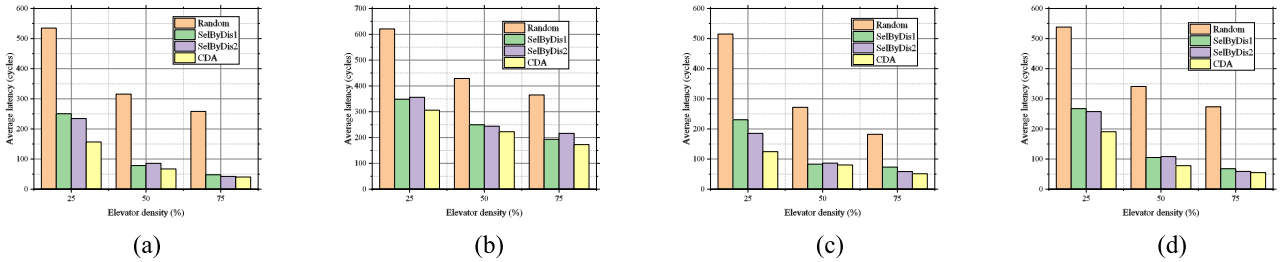


Fig. 17. Average network delay under real applications with different elevator densities. (a) Ferret. (b) Swaptions. (c) x264. (d) Freqmine.

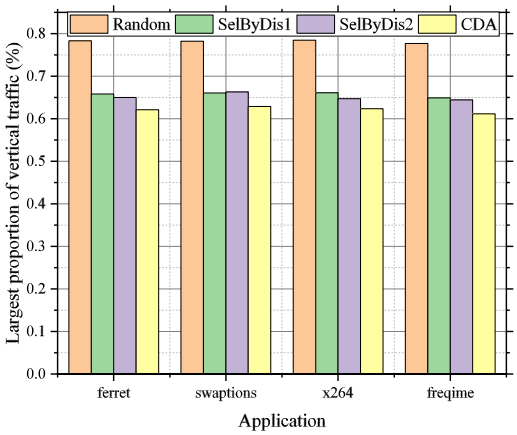


Fig. 18. Proportion of traffic loads of the elevator with the highest utilization in total vertical traffic.

Table IV, the improvement of the saturated injection rate of the proposed congestion-aware dynamic assignment (CDA) method compared with that of the corresponding method is shown in the brackets. For example, the random assignment method under the elevator density of 25% has a saturated injection rate of 0.014 packet/node/cycle. Compared with that, the proposed CDA method increases the saturated injection rate by 86%. In summary, the performance of the proposed CDA method is improved by 67%–86% compared with the random selection method, by 8%–25% compared with the SelByDis-1, and by 13%–18% compared with the SelByDis-2.

The comparison of saturated injection rates for different elevator assignment methods under uniform traffic can also be presented in Fig. 16. As shown in Fig. 16, the proposed CDA method has the highest saturated injection rate among the four assignment methods for the partially connected 3D-NoC. And the saturated injection rate of the fully connected 3D-NoC is presented with an orange dash line in the figure to show the performance comparison between the partially connected 3D-NoC and the fully connected 3D-NoC. It should be noted that we employ the *XYZ* routing in the fully connected 3D-NoC, in which the elevator located on the destination node is selected.

The results above are based on many synthetic traffic patterns. Besides, the proposed methods are also evaluated under real applications. First, we configured and ran several applications including ferret, swaptions, x264, and freqmine, which are all from the PARSEC benchmark [29], in the full system simulator Gem5 [24]. And we collected the traffic trace file from the Gem5 simulator. Then, we converted the trace file to the traffic table and feed it into the 3D-NoC simulator AccessNoxim. The average network delays under real applications with different elevator densities are shown in Fig. 17, and we found that the proposed CDA method has the best performance compared with other assignment methods.

C. Reliability Analysis

Heavy workload is considered to be one of the main reasons for TSV failures [30]. We have a balanced workload distribution among elevators and improve the mean-time-to-failure (MTTF) of the partially connected 3-D NoC significantly by using the proposed vertical link placement algorithm and the proposed dynamic elevator assignment scheme. Because the main reason for lowered MTTF is the high TSV utilization [30], the elevator with the highest utilization has the lowest MTTF. We use the proportion of traffic loads of the elevator with the highest utilization in total vertical traffic $(\max(T_{\text{elevator\_i}})/\sum T_{\text{elevator\_i}})$ to show the effect of these methods proposed in this article on their MTTF, which is shown in Fig. 18. As Fig. 18 shows, under four different real applications, the proposed CDA method can improve the MTTF of the partially connected 3D-NoC.

VI. Conclusion

In this article, we propose a hybrid strategy of the GA and the TS for optimizing elevator placement. The hybrid placement strategy can find multiple solutions in one single simulation run, so it is suitable for exploring the design space. However, static assignment methods can not dynamically change the elevator assignment according to the real-time state of the network, thus it may lead to network congestion. In this article, we propose a congestion-aware dynamic elevator assignment method. Experiments show that the performance of the CDA method is superior to the other three algorithms (random, SelByDis-1, SelByDis-2) in four traffic patterns including uniform, transpose, shuffle, and bit-reversal and four real applications from the PARSEC benchmark. Taking the uniform traffic pattern as an example, the performance of the CDA method is improved by $67\% - 86\%$ compared with the random selection method, by $8\% - 25\%$ compared with the SelByDis-1, and by $13\% - 18\%$ compared with the SelByDis-2. Besides, the proposed CDA method can improve the MTTF of the partially connected 3D-NoC. The hardware implementation results show that the area cost of the key component for the CDA method, the PSM, is negligible compared with a router.

REFERENCES

[1] K. J. Chen, C. Chao, and A. A. Wu, "Thermal-aware 3D network-on-chip (3D NoC) designs: Routing algorithms and thermal managements," *IEEE Circuits Syst. Mag.*, vol. 15, no. 4, pp. 45–69, 4th Quart., 2015.

[2] A. W. Topol *et al.*, "Three-dimensional integrated circuits," *IBM J. Res. Develop.*, vol. 50, nos. 4–5, pp. 491–506, Jul. 2006.

[3] V. Pavlidis and E. 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.

[4] A. Sheibanyrad, F. Ptrot, and A. Jantsch, *3D Integration for NoC-based SoC Architectures*, 1st ed. New York, NY, USA: Springer, 2011.

[5] J. Zhao, Q. Zou, and Y. Xie, "Overview of 3-D architecture design opportunities and techniques," *IEEE Design Test*, vol. 34, no. 4, pp. 60–68, Aug. 2017.

[6] D. Velenis, M. Stucchi, E. J. Marinissen, B. Swinnen, and E. Beyne, "Impact of 3D design choices on manufacturing cost," in *Proc. IEEE Int. Conf. 3D Syst. Integr. (3DIC)*, San Francisco, CA, USA, Sep. 2009, pp. 1–5.

[7] F. Dubois, A. Sheibanyrad, F. Pétrot, and M. Bahmani, "Elevator-first: A deadlock-free distributed routing algorithm for vertically partially connected 3D-NoCs," *IEEE Trans. Comput.*, vol. 62, no. 3, pp. 609–615, Mar. 2013.

[8] M. Bahmani, A. Sheibanyrad, F. Pétrot, F. Dubois, and P. Durante, "A 3D-NoC router implementation exploiting vertically-partially-connected topologies," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Amherst, MA, USA, Aug. 2012, pp. 9–14.

[9] J. Lee and K. Choi, "A deadlock-free routing algorithm requiring no virtual channel on 3D-NoCs with partial vertical connections," in *Proc. 7th IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, Tempe, AZ, USA, Apr. 2013, pp. 1–2.

[10] A. Kologeski, F. L. Kastensmidt, V. Lapotre, A. Gamatié, G. Sassatelli, and A. Todri-Sanial, "Performance exploration of partially connected 3D NoCs under manufacturing variability," in *Proc. IEEE 12th Int. New Circuits Syst. Conf. (NEWCAS)*, Trois-Rivieres, QC, Canada, Jun. 2014, pp. 61–64.

[11] E. Taheri, A. Patooghy, and K. Mohammadi, "Cool elevator: A thermal-aware routing algorithm for partially connected 3D NoCs," in *Proc. 6th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Mashhad, Iran, Oct. 2016, pp. 111–116.

[12] E. Taheri, M. Isakov, A. Patooghy, and M. A. Kinsy, "Advertiser elevator: A fault tolerant routing algorithm for partially connected 3D network-on-chips," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 136–139.

[13] Y. Fu *et al.*, "Congestion-aware dynamic elevator assignment for partially connected 3D-NoCs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, May 2019, pp. 1–5.

[14] S. Foroutan, A. Sheibanyrad, and F. Pétrot, "Assignment of vertical-links to routers in vertically-partially-connected 3-D-NoCs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 8, pp. 1208–1218, Aug. 2014.

[15] R. Salamat, M. Khayambashi, M. Ebrahimi, and N. Bagherzadeh, "A resilient routing algorithm with formal reliability analysis for partially connected 3D-NoCs," *IEEE Trans. Comput.*, vol. 65, no. 11, pp. 3265–3279, Nov. 2016.

[16] A. Charif, A. Coelho, M. Ebrahimi, N. Bagherzadeh, and N. Zergainoh, "First-Last: A cost-effective adaptive routing solution for TSV-based three-dimensional networks-on-chip," *IEEE Trans. Comput.*, vol. 67, no. 10, pp. 1430–1444, Oct. 2018.

[17] T. C. Xu, G. Schley, P. Liljeberg, M. Radetzki, J. Plosila, and H. Tenhunen, "Optimal placement of vertical connections in 3D network-on-chip," *J. Syst. Archit.*, vol. 59, no. 7, pp. 441–454, 2013.

[18] K. Manna, S. Chattopadhyay, and I. Sengupta, "Through silicon via placement and mapping strategy for 3D mesh based network-on-chip," in *Proc. 22nd Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Playa del Carmen, Mexico, Oct. 2014, pp. 1–6.

[19] K. Manna, S. Swami, S. Chattopadhyay, and I. Chattopadhyay, "Integrated through-silicon via placement and application mapping for 3D mesh-based NoC design," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 1, pp. 1–25, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/2968446

[20] R. Salamat, M. Khayambashi, M. Ebrahimi, and N. Bagherzadeh, "LEAD: An adaptive 3D-NoC routing algorithm with queuing-theory based analytical verification," *IEEE Trans. Comput.*, vol. 67, no. 8, pp. 1153–1166, Aug. 2018.

[21] O. Lysne, T. Skeie, S.-A. Reinemo, and I. Theiss, "Layered routing in irregular networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 1, pp. 51–65, Jan. 2006.

[22] F. Glover, J. P. Kelly, and M. Laguna, "Genetic algorithms and tabu search: Hybrids for optimization," *Comput. Oper. Res.*, vol. 22, no. 1, pp. 111–134, 1995, [Online]. Available: http://www.sciencedirect.com/science/article/pii/0305054893E0023M

[23] J. Lee, K. Kang, and K. Choi, "REDELF: An energy-efficient deadlock-free routing for 3D NoCs with partial vertical connections," *J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 3, pp. 1–22, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2751560

[24] N. Binkert *et al.*, "The Gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011. [Online]. Available: http://doi.acm.org/10.1145/2024716.2024718

[25] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, Jun. 2008.

[26] K.-Y. Jheng, C.-H. Chao, H.-Y. Wang, and A.-Y. Wu, "Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip," in *Proc. Int. Symp. VLSI-DAT*, Hsin Chu, Taiwan, Apr. 2010, pp. 135–138.

[27] F. Fazzino, M. Palesi, and D. Patti. (2008). *Noxim: Network-on-Chip Simulator*. [Online]. Available: http://sourceforge.net/projects/noxim

[28] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[29] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Dept. Doctor Philos., Princeton Univ., Princeton, NJ, USA, Jan. 2011.

[30] S. Das, J. R. Doppa, P. P. Pande, and K. Chakrabarty, "Reliability and performance trade-offs for 3D NoC-enabled multicore chips," in *Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE)*, Dresden, Germany, 2016, pp. 1429–1432.

Yuxiang Fu (Member, IEEE) received the B.S. degree in microelectronics and solid state electronics and the Ph.D. degree in electronic science and technology from Nanjing University, Nanjing, China, in 2013 and 2018, respectively.

In 2018, he joined the School of Electronic Science and Engineering, Nanjing University, where he is currently an Associate Research Professor. His current research interests include network-on-chip algorithms/architectures, low-power digital systems, and 3D IC design.

Chuan Zhang (Member, IEEE) received the B.E. degree in microelectronics and the M.E. degree in VLSI design from Nanjing University, Nanjing, China, in 2006 and 2009, respectively, and the M.S.E.E. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Minnesota (UMN) Twin Cities, Minneapolis, MN, USA, in 2012.

He is currently an Associate Professor with the National Mobile Communications Research Laboratory, School of Information Science and Engineering, Southeast University, Nanjing. His current research interests include 5G communication system designs, low-power high-speed VLSI design, specifically VLSI design for digital signal processing, digital communications (with emphasis on error-control coding and cryptography), quantum information theory, and biochemical synthesis implementation.

Wenqing Song (Student Member, IEEE) received the B.S. degree from the School of Electronic Science and Engineering, Southeast University, Nanjing, China, in 2017. She is currently pursuing the M.S. degree in electronic science and engineering with Nanjing University, Nanjing, China.

Her current research interests include polar coding algorithms and efficient hardware architecture.

Qinyu Chen (Student Member, IEEE) received the B.S. degree in communication engineering from Shandong University, Jinan, China, 2016. She is currently pursuing the Ph.D. degree with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China.

She is working on designing efficient hardware architectures to accelerate inference and training of deep neural networks (including convolutional neural networks, recurrent neural networks, and generative adversarial networks) with the goal of realizing power-efficient real-time inference and incremental learning of large networks on mobile devices.

Hui Chen (Graduate Student Member, IEEE) received the B.S. degree in electronic information science and technology from Jinling College, Nanjing University, Nanjing, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Electronic Science and Engineering.

His current research interests include digital integrated circuit design, VLSI computing IP optimization, and reconfigurable computing.

Minghao Zhou received the B.S. degree in electronic science and engineering from Nanjing University, Nanjing, China, in 2004, and the master's degree in electronic and information engineering from Shanghai Jiaotong University, Shanghai, China, in 2007. He is currently pursuing the Ph.D. degree in electronic science and engineering with Nanjing University.

His current research interests include 2.5D/3D system integration packaging technologies and thermal dissipation technology for ASIC and 3DIC.

Li Li (Member, IEEE) received the B.S. and Ph.D. degrees from Hefei University of Technology, Hefei, China, in 1996 and 2002, respectively.

She is currently a Professor of VLSI Design Institute, School of Electronic Science and Engineering, Nanjing University, Nanjing, China. Her current research interests include VLSI design for digital signal processing systems, reconfigurable computing and multiprocessor system-on-a-chip architecture design methodology.

Dr. Li is a member of Circuits & Systems for Communications TC of IEEE CAS Society.