3D Network-On-Chip Data Acquisition System Mapping Based on Reinforcement Learning and Improved Attention Mechanism

Chuanpei Xu
Xiuli Shi
Yang Wang (1033783706@qq.com)

Research Article

Keywords: data acquisition, 3D network-on-a-chip, NoC mapping, reinforcement learning, attention mechanism

Posted Date: July 12th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3142584/v1

Additional Declarations: No competing interests reported.

3D Network-On-Chip Data Acquisition System Mapping Based on Reinforcement Learning and Improved Attention Mechanism

Chuan pei Xu[1,2], Xiuli Shi[1,2],Yang Wang[1,2]

[1]School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China

[2]Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin 541004, China

Corresponding author: Yang Wang Email: 1033783706@qq.com Fax: +867732191029

Abstract

The 3D Network-on-Chip (NoC) data acquisition system utilizes NoC technology to establish a time-interleaved data acquisition system The mapping scheme determines the location of each Intellectual Property (IP) node in the NoC topology. The optimization of the mapping algorithm is one of the important means to reduce the communication delay of the acquisition system. The abundance of functional IP nodes in the 3D NoC data acquisition system creates a mapping challenge. To address this, we propose a mapping algorithm called Reinforcement Learning and an improved Attention Mechanism Mapping algorithm (RA-Map). The RA-Map mapping algorithm employs node function encoding and node position encoding to express the properties of an IP node in the task graph preprocessing. The local attention mechanism is used in the mapping network encoder, and the fusion of dynamic key node information is proposed in the decoder. The mapping result evaluation network achieves unsupervised training of the mapping network. These targeted improvements ultimately lead to an enhancement in mapping quality. Experimental results demonstrate that when compared to the discrete particle swarm algorithm and simulated annealing algorithm, the RA-Map mapping algorithm reduces the average communication cost by 6.5% and 8.5%, respectively. Furthermore, while ensuring mapping quality, it also shortens the mapping time.

Keywords: data acquisition; 3D network-on-a-chip; NoC mapping; reinforcement learning; attention mechanism

1. Introduction

With the rapid development of electronic information technology, the demand for data acquisition speed in electronic equipment status monitoring is increasing. Currently, System-on-Chip (SoC) faces bottlenecks such as low communication bandwidth and difficult clock synchronization, which affect the performance of the acquisition system[1]. To overcome these problems, NoC uses routing technology to replace bus structures for communication, thereby overcoming the limitations of traditional SoC. Data acquisition, storage, and transmission IP nodes are designed as resource nodes in a 3D NoC, achieving a high sampling rate and low latency time-interleaved data acquisition system[2].

The mapping scheme determines the location of each IP node in the NoC topology, and the mapping algorithm is one of the important means to achieve low latency in NoC[3]. Currently, mapping algorithms include exhaustive traversal and heuristic algorithms. For data acquisition systems, as the sampling rate increases and the number of IP nodes increases, exhaustive traversal becomes very time-consuming. Heuristic algorithms may get stuck in locally optimal solutions during iterative searches. NoC mapping can be viewed as selecting optimal decision variables in a discrete decision space, similar to the "action selection" in reinforcement learning in artificial intelligence[4]. Therefore, this paper proposes the RA-Map mapping algorithm, which consists of task graph preprocessing, mapping network, and mapping result evaluation network.

Based on the characteristics of the acquisition system task graph, this paper proposes an encoding method consisting of node function encoding and node position encoding to express all the properties of an IP node. In the encoder of the mapping network, a local attention operation is proposed based on the communication relationship between IP nodes to generate intermediate action vectors. In the decoder of the mapping network, a mapping environment is proposed that integrates global information, local information, and dynamic key node information of the task graph. Finally, the mapping result evaluation network is used to overcome the problem of difficult access to high-quality datasets and achieve unsupervised training of the mapping network.

The structure of this paper is as follows: Section 2 analyzes and discusses relevant work; Section 3 defines the task graph of the data acquisition system based on time-interleaved sampling and its mapping process; Section 4 proposes a data acquisition system mapping algorithm based on reinforcement learning and improved attention mechanism; Section 5 analyzes and compares mapping results; Section 6 summarizes this paper.

2. Related Works

For the small-scale NoC mapping problem with a small number of IP cores, several methods have been proposed, including Integer Linear Programming (ILP) and Branch and Bound algorithms. Ostler proposed a two-stage ILP-based strategy to map applications to complex multiprocessor, multithreaded network processors. This method effectively utilizes the parallel processing and multithreading capabilities of the target architecture, mitigating the high computational complexity of ILP [5]. Ghosh addressed the mapping problem on heterogeneous NoC platforms by minimizing energy consumption and maximizing system performance. They proposed a unified approach using mixed integer linear programming and random rounding instead of solving these subproblems sequentially. Although this method can obtain the optimal solution, its practical complexity is high [6]. Huang introduced an energy-aware and task allocation mapping algorithm based on heterogeneous multiprocessor systems. This algorithm extends the ILP formulation by explicitly considering the trade-off between processor and communication power consumption. To address the long execution time, the author optimized the mapping algorithm using simulated annealing, but the quality of the mapping solution was compromised [7]. Tosun proposed a clustering-based mapping method to map applications onto NoC architectures. This method divides tasks into small regions or grids called clusters, which are mapped onto smaller topological grids. The final result is obtained by combining all the smaller grids. Compared to ILP-based methods, this method has a faster execution speed but may increase communication costs [8]. Aravindhan introduced the KL algorithm to optimize the clustering-based mapping method. Experimental results showed that this method reduced communication costs and partitioning degree compared to the clustering method, improving the solution speed [9]. Lin proposed a new network interface and traffic balancing mapping algorithm based on an improved Branch and Bound algorithm. This method disperses traffic load to avoid router congestion caused by high traffic IP cores, but it may lead to high communication costs [10]. Reshadi proposed the Elixir algorithm, a minimum power mapping algorithm based on bandwidth constraints and Branch and Bound method. It consists of two steps: the first step calculates the mapping with the minimum communication cost using the concept of a search tree, and the second step selects the best mapping with minimum power consumption and delay using the Polaris toolchain. Compared to the original Branch and Bound algorithm, this method reduces both network delay and power consumption [11]. Khan proposed a multi-objective segmented brute-force mapping algorithm based on bandwidth constraints. This algorithm divides the application into multiple segments to achieve efficient mapping of embedded applications on NoC system processing units. However, it has a longer execution time [12].

The above exact mapping algorithms have high time complexity and are inefficient for large-scale NoC mapping problems. When solving large-scale NoC mapping problems, heuristic search algorithms are more efficient as they can find relatively optimal solutions within a limited time. Chatterjee proposed a reliability model that considers the thermal effects of computing and communication nodes to mitigate hotspot formation. The author implemented this model using both mixed integer linear programming and Particle Swarm Optimization (PSO) methods. Experimental results showed that the PSO-based heat reliability model can handle large-scale mapping problems and overcome the convergence issue faced by ILP methods for large-scale mapping [13]. Upadhyay proposed a mapping algorithm based on a reconfigurable architecture, which reduces communication costs for different applications using a two-stage PSO algorithm. The first stage combines multiple applications to achieve global mapping, and the second stage uses multiplexers to switch IP cores to nearby routers for reconfiguration. Although this method has a longer execution time, it reduces the system's communication costs [14]. Seidipiri proposed a mapping method that sorts IP cores based on their communication requirements. This method maps high-traffic nodes to the central part of the grid topology and further optimizes the mapping results by repeating this process. Experimental results showed that this method can reduce network communication costs, power consumption, and network latency [15]. Sahu proposed a performance-aware mapping algorithm based on Particle Optimization. Compared to ILP-based mapping algorithms that are close to the optimal solution, this method effectively reduces communication costs and improves both system energy consumption and performance [16]. Tosun proposed a low-complexity heuristic algorithm that maintains a priority list of tasks based on communication bandwidth and selects the initial IP core to be mapped based on this priority list. This process is repeated during mapping, and the priority list is updated after each mapping. Finally, the best mapping solution is selected from a set of mapping solutions as the final application mapping solution [17]. These heuristic algorithms provide solutions for large-scale NoC mapping problems, but their performance is closely related to their ability to avoid local optima..

In recent years, reinforcement learning methods have provided new research perspectives for NoC mapping. Some researchers have used reinforcement learning methods to solve NoC mapping problems. Chen proposed a mapping algorithm based on the combination of Graph Neural Networks and Pointer Networks [17]. However, the quality of the mapping solution of this algorithm is limited by the quality of the training dataset. Therefore, the author used Reinforcement Learning (RL) methods to train the mapping model and improve the quality of the solution [18]. However, due to the long short-term memory network used as the encoder in the mapping algorithm, there is a problem of long sequence forgetting. When dealing with large-scale NoC mapping problems, the quality of the mapping solution is limited. Therefore, other heuristic algorithms are used for secondary optimization of the mapping solution. Jagadheesh was inspired by neural network solutions to the Traveling Salesman Problem (TSP) and proposed a NoC mapping algorithm based on the actor-critic structure. Additionally, the 2-opt local search algorithm was used to improve the quality of the mapping sequence [19].

These algorithms provide different approaches and solutions for NoC mapping problems. The time complexity of exact mapping algorithms, such as ILP and Branch and Bound, is high. Heuristic algorithms may get trapped in local optima when solving large-scale mapping problems. Existing deep learning mapping solutions have not been optimized for specific mapping problems. This paper aims to optimize the mapping problem of high-speed data acquisition systems using reinforcement learning methods to improve the quality of the mapping solution.

3. Mapping problem definition for NoC data acquisition systems

3.1 Time interleaved data acquisition system architecture

The time interleaved data acquisition system can be divided into four parts according to functions: data acquisition, memory, transmission interface, and acquisition system controller. In Figure 3-1, the acquisition system controller is responsible for clock calibration and system cooperative control; the data acquisition ADC is controlled by the system controller to acquire data; the memory is used to realize data storage and forwarding; the transmission interface is responsible for uploading the acquired data and receiving commands from the host computer.

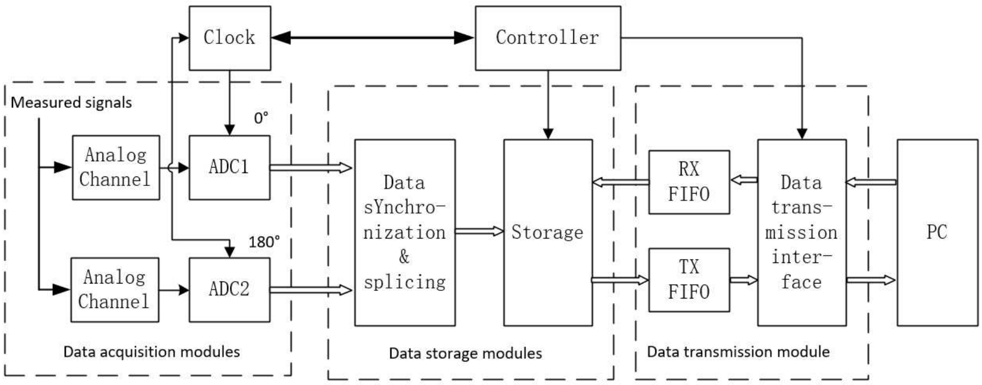

Figure 3-1 Block diagram of time interleaved data acquisition system

Based on the temporal interleaved data acquisition system model, all IP cores in the system are abstracted as task nodes.The data communication relationships between IP cores are then mapped as connection relationships between task nodes. This approach is used to build the task graph of the NoC data acquisition system to support the optimization of the data acquisition system. The task diagram of the data acquisition system is in Figure 3-2. In the acquisition system based on the time interleaving technique, all acquisition nodes have the same sampling rate, but the time when each node starts sampling varies periodically. The data acquisition process has the same amount of communication, data transmission bandwidth requirements, and latency limits between each acquisition node and storage node.

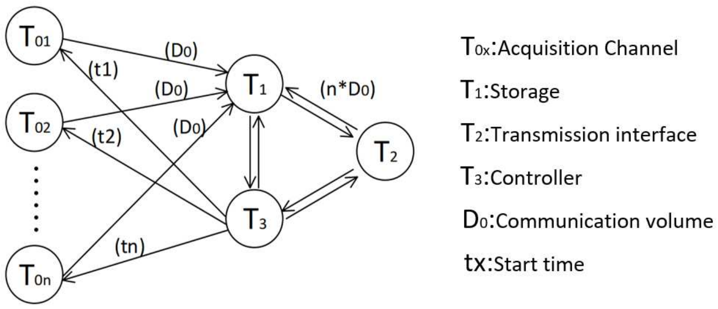

Figure 3- 2 Data Acquisition System Task Map

3.2 NoC data acquisition system mapping architecture

The task graph to be mapped is a directed graph $G\ (C, W)$ where each vertex $c_i \in C$ represents an IP core and the edge $w_{i,j} \in W$ represents IP cores c_i to the IP core c_j the amount of data communication.

Target NoC topology $A(R, L)$ with each vertex $r_k \in R$ denotes a routing node and the edge $l_{k,l} \in L$ denotes a routing node r_k to the routing node r_l communication, the 3DMesh NoC is in Figure 3-3.

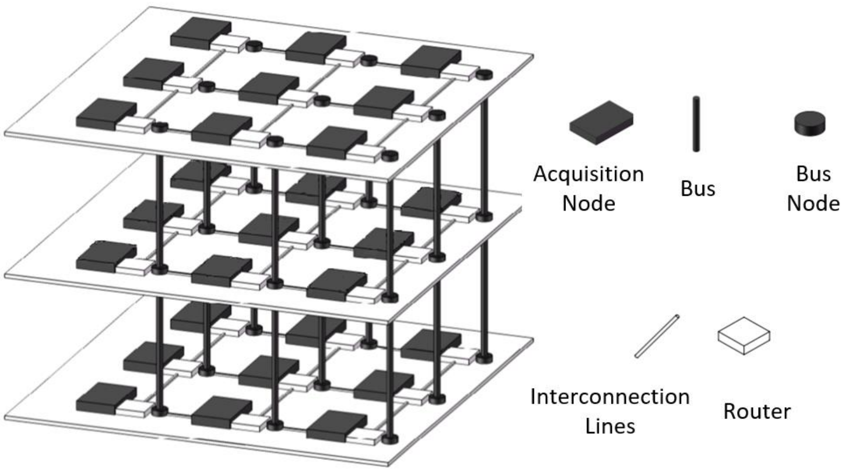

Figure 3-3 3D Mesh NoC

Task map to be mapped $G(C, W)$ to the target NoC $A(R, L)$ The mapping is defined by the mapping function f defined by $C \rightarrow R$, for example $\forall c_i \in C$, and $\exists r_k \in R$ and $f(c_i) = r_k$ and $\forall c_i \neq c_j$, the mapping $f(c_i) \neq f(c_j)$. The mapping function f is a probabilistic model that, given a task mapG under the probability distributionp $_\theta$ $(M|G)$ from which the mapping solution is sampled.

IP core to topology router is a one-to-one mapping that satisfies $|C| \leq |R|$. If$|C| < |R|$, then add $|C| - |R|$ a virtual IP core to the task graph to satisfy$|C| = |R| = n$[20]. The IP core mapping can be described by the following equation:

$$\forall c_i \in C,\ f(c_i) = r_k \in R \tag{3-1}$$

$$\forall c_j \in C,\ f(c_j) = r_l \in R \tag{3-2}$$

$$\forall c_i \neq c_j, \forall r_k \neq r_l \tag{3-3}$$

$$|C| = |R| = n \tag{3-4}$$

The mapping solution reflects the mapping of IP cores in the task graph to router node locations in the network topology. Assuming that the indexes of the network topology are numbered hierarchically from top left to bottom right, the mapping solution can be expressed as from1 to n in a numerical arrangement $M = \{m_i\}_{i=1}^{n}$, where each element represents an IP core in the task graph and the index, represents a router node on the corresponding network topology. This numerical arrangement forms the mapping solution sequence, which corresponds to the router arrangement in the NoC topology. In Figure 3-4, a task graph of a data acquisition system containing eight IP cores is mapped onto a 3D Mesh structure, where a set of optimal mapping solutions are $M = \{1, 3, 2, 4, 8, 6, 5, 7\}$, where the second element is 3, indicating that the second IP core in the task graph is mapped onto the third router node in the NoC topology.

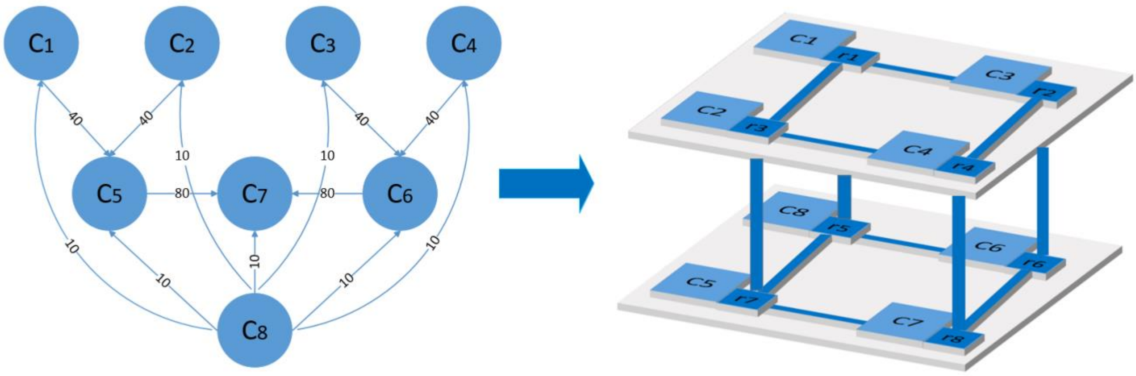

Figure 3-4 Mapping results of eight-node data acquisition system

3.3Evaluation metrics for NoC data acquisition systems

The evaluation metrics of NoCs are latency, throughput rate, etc. Both latency and throughput rate in data acquisition systems are related to the distance between any two functional nodes with data communication in the NoC architecture[21-23], so the mathematical model for the communication of NoC data acquisition systems in 3D Mesh architecture is defined as follows:

$$t = t_a + t_l + t_s + t_w \tag{3-5}$$

wheret_a is the time taken by the resource node to transmit the data to the network;t_l is the time it takes for the data to pass through all channels; t_s means the time to be spent on the communication nodes; t_w refers to the blocking time.

From equation (3-5), it can be seen that the delay of the communication task is closely related to the chosen communication path. To facilitate the calculation of the communication delay, it is assumed that when communicating, the path with the shortest distance between two nodes in the network is always selected. Using $c_i = (x_s, y_s, z_s)$ and $c_j = (x_d, y_d, z_s)$ denote the coordinates of the source and target nodes, respectively, and based on the XYZ routing algorithm[24] it is known that such that $d_x = |\Delta x| = |x_s - x_d|$, $d_y = |\Delta y| = |y_s - y_d|$ and $d_z = |\Delta z| = |z_s - z_d|$, then the number of hops between the source routing node and the destination routing node is :

$$hop_{i,j} = d_x + d_y + d_z \tag{3-6}$$

Assuming that all available channels are idle, the transmission time of the shortest path is given by $t_l =$

$hop_{i,j} \times (w_{i,j}/s) \times t_1$. where the constant $w_{i,j}$ is the number of c_i node to c_j the amount of data transferred from node to node, and a fixed value then denotes the size of a unit information packet, and t_1 denotes the time required for a unit information packet of data to pass through a single channel and communication node when the channel and communication node are in the idle state.

If only the delay model under congestion-free conditions is considered, due to s and t_1 being constants, the transmission delay of the shortest path is proportional to the number of transmission hops. Therefore, the $t_l \approx hop_{i,j} \times w_{i,j}$.

The overall communication latency of the data acquisition system is

$$T = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} \times hop_{i,j} \tag{3-7}$$

According to equation (3-7), it is known that the communication delay of the data acquisition system is related to the communication volume of the overall acquisition system and the number of routing hops between communication nodes after mapping. As in equation (3-8), the communication cost independent of the communication volume of the acquisition system is obtained by normalizing the overall communication volume of the acquisition system for different acquisition data task graphs T_{norm}, the communication delay of the acquisition system is positively correlated with the communication cost. The communication cost represents the average number of communication hops between any two nodes and its ideal value is 1. In this paper, the T_{norm} is the objective function for optimization.

$$T_{norm} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} \times hop_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{n} hop_{i,j}} \tag{3-8}$$

4. Mapping algorithm based on reinforcement learning and improved attention mechanism

4.1 The mapping scheme for the 3D NoC data acquisition system

The essence of the mapping problem is the NP-hard problem, when the number of IP nodes to be mapped is n, there are n! number of solutions[24]. In NoC data acquisition systems, as the number of data acquisition ADCs and matching memories increases, the number of IP cores involved in the mapping problem increases, the solution space of the problem is huge, and it is difficult to obtain the optimal solution quickly, and the training set composed of the best instances is also difficult to obtain. Reinforcement learning does not require the training set composed of the best instances, but the optimal mapping strategy is finally obtained through continuous trial and error. First, the NoC mapping process is modeled as a Markov process, where the state s of the NoC mapping consists of the input sequence of task graph nodes and the mapped nodes, and the actions are the nodes selected in t step a_t, and all actions in sequence form the solution of the NoC mapping problem. The reward r is the negative value of the communication cost of the data acquisition system, the communication cost of the data acquisition system needs to be minimized. The policy is then the mapping of state s to action a, the probability of selecting the node to be mapped $p_\theta(a|s)$:

$$p_\theta(a|s)=\prod_{t=1}^{n} p_\theta(a_t|s,a_{1:t-1}) \tag{4-1}$$

The NoC data acquisition system mapping strategy is represented by a neural network with parameters θ The neural network is represented by the probability of each action step as $p_\theta(a_t|s,a_{1:t-1})$, the probability of selecting the node to be mapped based on the mapped nodes $a_{1:t-1}$ and the input node sequence s to calculate the probability of selecting the node to be mapped. The sequence of input nodes can be obtained according to the chain rules to the final mapped node sequence π The mapping probability of $p_\theta(\pi|s)$ Therefore, a neural network model with parameter θ A neural network model with parameters[18] can be built to represent this mapping strategy.

The reinforcement learning algorithm in which the intelligence continuously performs actions until the end of the round. After a round ends, the intelligence calculates the total feedback reward for that round and then uses that reward to update the parameters of the policy. the total feedback reward in the NoC mapping problem is the negative value of the communication cost of the data acquisition system. Reinforcement learning is based on equation (4-2) to calculate and update the gradient of the mapping strategy to obtain the appropriate parameters θ and thus the solution to the NoC mapping problem.

$$\begin{gathered}\nabla\mathcal{L}(\theta|s)=E_{p_\theta(a|s)}[(L(a|s)-b(s))\nabla\ln p_\theta(a|s)],\\ \theta\leftarrow\theta+\nabla\mathcal{L}(\theta|s)\end{gathered} \tag{4-2}$$

Calculating the mapping model parameters according to the chain rule θ the partial derivative of the gradient value can be obtained $\nabla\ln p_\theta(\pi|s)$, and $(L(\pi|s)-b(s))$ This gradient value determines the direction of gradient descent. Where $L(a|s)$ is the performance of the current strategy, and $b(s)$ reflects the average performance of the strategy. If the current strategy performs better than the "average", it is motivated in the positive direction, and vice versa. For this purpose, a Critic neural network is introduced $b(s)$ to predict the communication cost of the data collection task graph to be mapped.

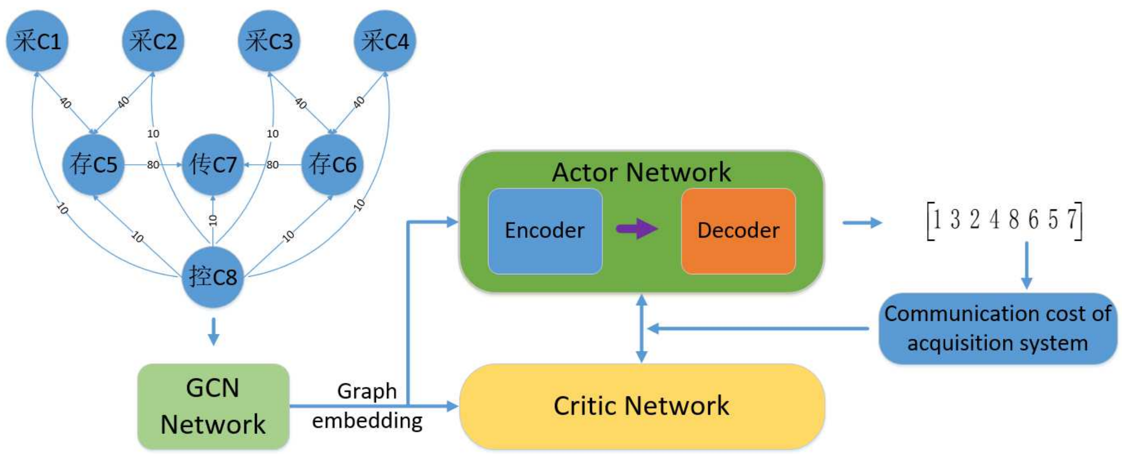

Figure 4-1 Block diagram of RA-Map data acquisition system mapping algorithm

In this thesis, an NoC mapping algorithm RA-Map based on reinforcement learning[25] and an improved attention mechanism[26] is proposed, In Figure 4-1. The algorithm contains three parts: task graph preprocessing, mapping network (Actor), and mapping result evaluation network (Critic). The task map preprocessing uses the graph convolution method to extract node features and graph structure features in the task map to generate node embedding vectors. The mapping network consists of two parts, the encoder, and the decoder. The encoder performs local attention operations on the task graph node embedding vectors to generate node action vectors. The decoder combines the task graph global information, local information, and key node information as the current environment state, and calculates the probability of each unmapped node as the next mapped node based on the mapping policy to obtain the mapping solution sequence and calculate the communication cost of the data acquisition system. The higher the probability of the mapping solution sequence, the lower the overall communication cost[19]. In addition, the Critic network is introduced to estimate the communication cost of the data acquisition system and guide the Actor network to update the mapping strategy.

4.2 Task map pre-processing

Task graph preprocessing generates node embedding vectors. In the task graph of the data acquisition system, each functional node consists of two parts: node function coding and node position coding. For example, in the data acquisition system, one-hot coding is used to classify the node functions as acquisition node [1, 0, 0, 0], transmission node [0, 1, 0, 0], storage node [0, 0, 1, 0] and control node [0, 0, 0, 1]. However, with such an encoding method, the encoding results of functional nodes of the same type are consistent, and it is impossible to distinguish the different numbers of each functionally identical node. Therefore, to distinguish the same type of functional nodes, an encoding (Pos) representing the location information of each functional node is added to the one-hot encoding of that node. The encoding method of Pos adopts the absolute position encoding method, and its expression is:

$$\mathrm{Pos_i(t)} = \sin\left(\frac{\mathrm{t}}{100^{\mathrm{i/d}}}\right), \mathrm{i} \in \mathrm{d} \tag{4-3}$$

where the symbol t represents the ordinal number of a functional node in the task graph. The symbol d is the dimension of the position encoding vector, and in this paper, four elements of the vector are used to represent the position information of a functional node. The symbol i denotes the ith element in the position vector, the encoding value of the ith dimension. Specifically, taking the acquisition nodec_3 As an example, the acquisition nodec_3 is coded as [1, 0, 0, 0] and the location is coded as [0.812, -0.988, 0.095, 0.030], so the full encoding of the acquisition nodec_3 of the complete code is [1, 0, 0, 0, 0, 0.812, -0.988, 0.095, 0.030].

First, the one-hot encoding vector of each functional node is stitched with the position encoding vector to form the node feature vector $\mathrm{X} = (\mathrm{x_1}, \cdots, \mathrm{x_n}) \in \mathbb{R}^{\mathrm{n}\times 8}$. Then, the node feature vector is used as the initial node feature and the initial node feature matrix and the task graph adjacency matrix are used as inputs. Finally, the connections between nodes are captured by a 3-layer Graph Convolutional Network (GCN)[27] to obtain a more comprehensive embedding representation of each node in the task graph $\mathrm{E} = (\mathrm{e_1}, \cdots, \mathrm{e_n}) \in \mathbb{R}^{\mathrm{n}\times \mathrm{d_e}}$ to be used as input for the subsequent mapping model.

4.3 Actor Network

The mapping network completes the mapping of IP cores and is a typical sequence input to sequence output (Seq2Seq) structure. In Figure 4-2, the input of the encoder consists of the embedding vector of the task graph nodes $\mathrm{E} = (\mathrm{e}_1, \cdots, \mathrm{e}_\mathrm{n}) \in \mathbb{R}^{\mathrm{n} \times \mathrm{d}_\mathrm{e}}$ and the adjacency matrix of the task graph $\mathrm{D} \in \mathbb{R}^{\mathrm{n} \times \mathrm{n}}$. The encoder converts each node in the task graph into an intermediate action vector $\mathrm{A} = (\mathrm{a}_1, \cdots, \mathrm{a}_\mathrm{n}) \in \mathbb{R}^{\mathrm{n} \times \mathrm{d}_\mathrm{a}}$ which is used as the input of the decoder. The decoder generates a sequence of mapping solutions based on the intermediate action vectors according to the mapping probability model.

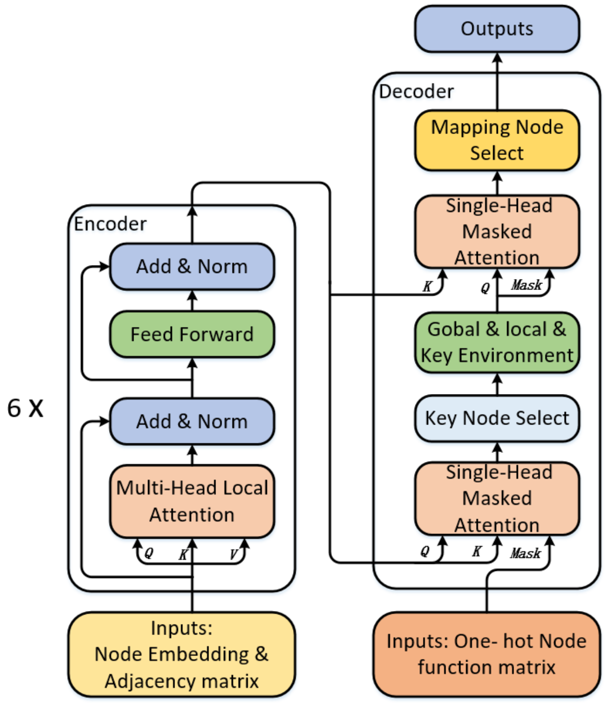

Figure 4-2 Block diagram of RA-Map data acquisition system mapping algorithm

4.3.1 Encoder

The mapping network encoder retains the useful part of the input information and generates intermediate action vectors. In a data acquisition system, the data flow between nodes follows the sequence of data collection, storage, and forwarding, and there is no direct connection between two nodes without data transmission. Therefore, to avoid introducing useless information in the intermediate action vector A, the encoder uses a multi-head local attention mechanism to generate the intermediate action vector. The structure of the encoder is shown in Fig. and consists of six identical blocks, each of which contains a multi-head local attention layer and a feed-forward fully connected layer. In addition, the blocks are connected in the form of residual connections to further improve the performance and stability of the model.

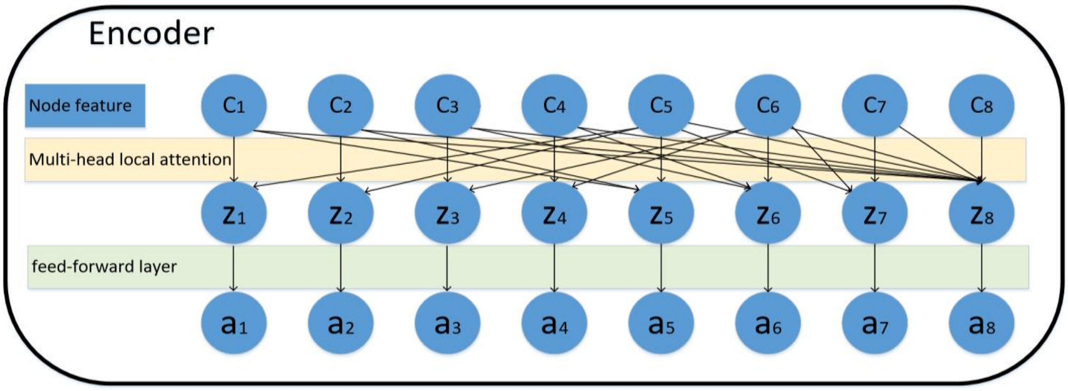

Figure 4-3 Encoder structure

In the local multi-head attention operation, the Mask matrix reflecting the connection relationship between nodes is obtained based on the adjacency matrix of the task graph, which is calculated as follows:

$$\begin{cases} x = 1, & x > 0 \\ x = 0, & x \leq 0 \end{cases} \tag{4-4}$$

For each node in the task graph, the multi-head local attention operation aggregates the information of that node and its neighbor nodes and calculates the intermediate action vector of that node by weighted summation A . where the weighting factor is determined by both the relationship between nodes and the Mask matrix. Specifically, a position of 1 in the Mask matrix indicates the existence of a connection relationship between two nodes, while a position of 0 indicates the absence of a connection relationship between two nodes. The local multi-head attention operation is calculated as follows:

$$A = \mathrm{softmax}\left(\frac{QK^T \odot \mathrm{Mask}}{\sqrt{d}}\right) V \tag{4-5}$$

where $Q = (q_1, \cdots, q_n)$,$K = (k_1, \cdots, k_n)$, and $V = (v_1, \cdots, v_n)$ denote the matrices of query, key and value respectively. Specifically, each element of the input sequence is represented as a vector. For each element, three weight matrices are constructed W_q , and W_k and W_v and mapping the element vectors to the new query vector, by matrix multiplication Q , the key vector$K$, and the value vector K respectively. When calculating the attention score, the query vector Q and key vector K matrix multiplication are performed to obtain the similarity score of each query vector with all key vectors. The similarity scores are multiplied with the corresponding elements of the Mask matrix to determine which similarity scores should be set to zero, which in turn is used as a weighting factor to calculate the weighted sum of the value vectors.

The output of the multi-head local attention operation is normalized by the feed-forward fully connected layer with:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{4-6}$$

Where x is the output of the multi-head local attention operation, the W_1 and b_1 are the weight matrix and bias vector of the first linear transformation, $\max(0, X)$ is the ReLU activation function, and W_2 and b_2

are the weight matrices and bias vectors of the second linear transformation.

4.3.2 Decoder

The mapping network decoder implements the mapping of IP cores. In the data acquisition system, the storage node plays a top-down role and is responsible for receiving data from the acquisition node on the one hand and forwarding the stored data to the transmission node on the other hand. Therefore, the storage node is the key node in the data acquisition task graph. To address this feature, in the decoder, the task graph key information is introduced to solve the mapping sequence based on considering the task graph's global information and local information. At the $t = 4$ step, the structure of the decoder is shown in Fig 4-4.

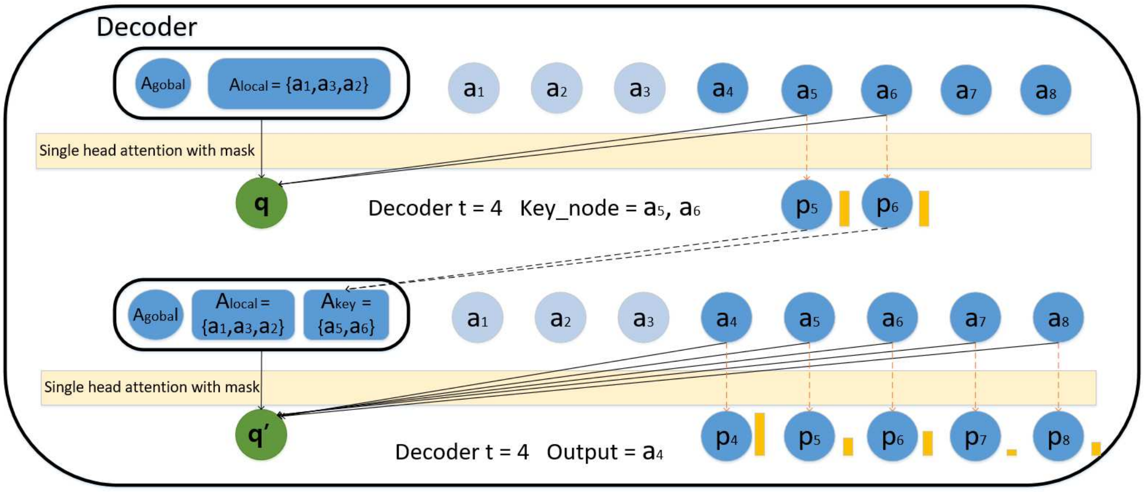

Figure 4-4 Decode structure

In the decoder, the intermediate action vector output by the encoder $A = (a_1, \cdots, a_n) \in \mathbb{R}^{n \times d_a}$ and the node feature vector $X = (x_1, \cdots, x_n) \in \mathbb{R}^{n \times 4}$ as inputs. In which, the global information of the task graph is obtained by averaging the intermediate action vectors A_{gobal} and the intermediate action vectors of the three most recently mapped nodes are used as the local information of the task graph A_{local} When the number of mapped nodes is less than three, the global information is obtained. When the number of mapped nodes is less than three, a mapping model learnable occupancy vector is introduced a_0. The global and local information of the task graph is stitched together and transformed by the fully connected layer to obtain ad dimensional Query vector. Among all the stored nodes, the two stored nodes with the largest probability values are dynamically selected as the key information of the task graph by the single-head attention operation with mask, which is calculated as

$$a_i = QA^T \odot Mask \,, \quad i \in \{1 \cdots n\} \tag{4-7}$$

$$p_i = softmax(a_i) = \frac{\exp(a_i)}{\sum_{j=1}^{N} \exp(a_j)} \,, \quad i \in \{1 \cdots n\} \tag{4-8}$$

where Q denotes the Query vector obtained by aggregating the global and local information of the task graph, A is the intermediate action vector, and the Mask matrix is generated from the node feature matrix.$\odot$ denotes the corresponding elements of the matrix are multiplied, andp_i denotes the probability of each stored node.

In the decoder, the global information of the task graph is first splicedA_{gobal} , local informationA_{local} and dynamic key node information A_{key} and then a fully connected layer transformation generates a d dimensionalQuery$'$ vector. The action encoding vectors of unmapped nodes are used as Key vector, and the probability of each unmapped node being the next mapped node is calculated using a single-head attention mechanism with a mask P, and at the $t = 4$ at the step $P_{t=4} = \{p_4, p_5, p_6, p_7, p_7\}$. In the model training process, a polynomial sampling method is used to sample each node to be mapped according to the probability P sampling, the higher the probability that the unmapped node with a higher probability value is selected as the next mapped node, thus enhancing the diversity of model mapping results. When testing the model, the unmapped node with the highest probability value is selected as the next mapped node to achieve the best results. In the figure, the action vector a_4 has the highest probability value, so the acquisition node c_4 is the next mapping node.

The decoder is autoregressive and achieves a complete mapping of task graph nodes by cyclically executing the decoding process. When all nodes are finished mapping, a sequence of mapping solutions is obtained $M = \{m_i\}_{i=1}^{n}$[18] , which is a numerical arrangement from 1 to n, where the index indicates the routing nodes on the architecture and the elements indicate the IP cores in the task graph. Finally, the communication cost of this solution sequence is calculated by T_{norm} to evaluate the performance of the mapping network.

4.4 Critic Network

The Critic network predicts the communication cost for a given graph of data collection tasks to be mapped. In the training phase, if the mapping network generates a sequence of mapping solutions M the communication cost of T_{norm} is smaller than the communication cost estimated by the Critic network $T_{norm\_estimate}$. The input to the Critic network is a task graph node embedding vector obtained by graph convolution preprocessing $E(e_1, \cdots, e_n)$, similar to the Actor network, which employs a three-level multi-head attention block to extract sequence features. Each attention block consists of a multi-head attention sub-layer and a feed-forward fully connected sub-layer, using IN to normalize the node sequences output from each layer. In addition, the network layers are connected to each other using residual networks to prevent gradient disappearance or explosion. Then, all node information in the task graph to be mapped is averaged. Finally, a two-layer linear transformation is used to obtain the predicted value of the communication cost of the data acquisition system to be mapped.

4.5 Model Training

Monte Carlo sampling is a random sample-based estimation method that can be used to approximate the expectation in the strategy gradient formula, thus avoiding the complexity and difficulty associated with the direct calculation of the expectation. Therefore, Monte Carlo sampling is used to approximate the strategy gradient formula (3-10) for effective strategy optimization in reinforcement learning[19].

$$\nabla\mathcal{L}(\theta|s) \approx \frac{1}{B}\sum_{i=1}^{B}(L(a_i|s_i) - b(s_i))\nabla \ln p_\theta(a_i|s_i) \tag{4-9}$$

where $\frac{1}{B}\sum_{i=1}^{B}\big(L(a_i|s_i) - b(s_i)\big)$ denotes the actual communication cost of the mapping network solution sequence at the end of a round T_{norm} and the critic predicted communication cost$T_{norm_estimate}$ which determines the direction of gradient descent. The reinforcement learning algorithm calculates the mapping network by equation (4-2) $P_\theta(a_i|s_i)$ parameters in θ and implements updates to obtain a more accurate mapping strategy.

The Critic network is trained simultaneously with the NoC mapping network (Actor network), where the strategy network outputs the mapping solution sequence communication cost as the actual value and Critic's estimated communication cost as the predicted value, and the Critic network is trained by minimizing the mean square error between the actual and estimated values to improve its prediction accuracy. In this way, the Critic network can more accurately evaluate the performance of the mapping solution sequences generated by the policy network and thus guide the update of the policy network.

RA-Map uses the steps in Algorithm 1 to train the parameters in the network. The training dataset was randomly generated by the data acquisition system model containing 8, 12, 27, 31, 46, 50, 73, 106, and 124 IP nodes constituting the task maps of the data acquisition system, and a total of 25,600 different task maps were generated as the training set. In the model training phase, the mapping network uses Adam optimizer[28] and learning rate 1e-5 to tune the network parameters and a gradient cropping strategy to limit the magnitude of the gradient to 1.0 to avoid the problem of gradient explosion. The Critic network was trained using the Adam optimizer and a learning rate of 1e-3. The whole training process took about 43 hours.

Algorithm 1: RA-Map data acquisition system mapping algorithm

Input: Task graph

Output: Trained model

Training procedure:

1 Initialize training set S.

2 number of epochs e.

3 batch size B.

4 Initialize actor network parameters θ

5 Initialize critic network parameters φ

for epoch $= 1$ to e **do**

Generate B Task Graph $\{s_i\}_{i=1}^{B}$

Sample a mapping solution π_i for each Task graph s_i, calculate communication cost $L(\pi_i)$, obtain$\{L(\pi_i)\}_{i=1}^{B}$

Calculate a base function $b(s_i)$ for each Task graph s_i, obtain $\{b(s_i)\}_{i=1}^{B}$

$\nabla\mathcal{L}(\theta|s) \approx \frac{1}{B}\sum_{i=1}^{B}(L(\pi_i|s_i) - b(s_i))\,\nabla \ln p_\theta(\pi_i|s_i)$

$\theta \leftarrow \mathrm{Adam}(\theta + \nabla\mathcal{L}(\theta|s)\ )$

$\varphi \leftarrow \mathrm{Adam}(d\varphi\ )$

end for

5. Experimental results and discussion

The RA-Map data acquisition system mapping algorithm is programmed in Python 3.9, using a third-party machine learning framework, PyTorch 1.11. Simulation validation is performed based on an Intel Gold 6258R CPU, 192GB of RAM, and an NVIDIA Quadro RTX 8000 GPU.

5.1 Experimental procedure and validation results

According to the data acquisition system model, the task map of the data acquisition system with the number of IP nodes 8, 12, 27, 31, 46, 50, 73, 106, 124 is randomly generated and used as a benchmark to test the effectiveness of the proposed RA-Map algorithm. In Table 5-1, the changes in the communication cost of the mapping solution sequence are compared for the RA-Map mapping algorithm before and after the encoder and decoder improvements. While keeping the other parts of the mapping algorithm unchanged, changes are made to two aspects: change 1 is to replace the multi-head local attention operation in the encoder part of the RA-Map algorithm with a multi-head global attention operation; change 2 is to remove the dynamic key node information fused in the decoder part of the RA-Map algorithm and retain only the global and local information of the task map as the current environment to complete decoding.

Table 5-1 Comparison results of the RA-Map mapping algorithm before and after improvement

IP cores	RA-Map		RA-Map(No Local Att-Encoder)			RA-Map (No key-Decoder)		
	T_{norm} (hop)	Time (s)	T_{norm} (hop)	Cost normalized to RL-MAP	Time (s)	T_{norm} (hop)	Cost normalized to RL-MAP	Time (s)
8	1.294	0.53	1.294	1.0	0.53	1.294	1.0	0.53
12	1.375	0.62	1.375	1.0	0.61	1.375	1.0	0.62
18	1.293	2.18	1.426	1.103	2.1	1.304	1.009	2.18
27	1.368	5.39	1.652	1.208	5.14	1.403	1.026	5.36
31	1.345	6.47	1.949	1.449	6.07	1.458	1.084	6.42
46	1.594	9.64	2.173	1.363	9.21	1.789	1.122	9.58
50	1.754	10.04	2.279	1.299	8.86	1.983	1.131	9.081

73	1.921	12.58	2.739	1.426	11.94	2.368	1.233	12.31
106	2.106	20.56	2.869	1.362	19.74	2.468	1.172	20.27
124	2.508	28.28	3.368	1.343	27.31	2.964	1.182	27.85
		Average		1.255			1.096	

In Table 5-1, when the number of IP cores in the data acquisition system is less than or equal to 12, there is no difference in the communication cost of the mapping solution sequence before and after the improvement of the RA-Map mapping algorithm. However, when the number of IP cores is greater than 12, and when the number of IPs is higher, the use of the multi-head local attention mechanism in the encoder and the inclusion of dynamic key node information in the decoder both help to improve the quality of the model mapping solution. Overall, based on benchmark tests with different numbers of IP cores in 10 groups, the RA-Map mapping algorithm reduces the average communication cost by 25.5% compared to the mapping model before the improvement of the encoder coder, and the RA-Map mapping model reduces the average communication cost by 9.6% compared to the decoder mapping model without incorporating dynamic key node information. algorithm, there is almost no difference in running time, so it is not discussed separately.

To evaluate the performance of the proposed RA-Map algorithm, it is compared with the particle swarm algorithm and simulated annealing algorithm of the heuristic algorithm on the 3D NoC data acquisition system mapping problem. Table 5-2 shows the results of the proposed RA-Map algorithm compared with other heuristic algorithms in terms of mapping solution communication cost and running time based on benchmark tests with different numbers of IP cores in 10 groups.

Table 5-2 Comparison results between RA-Map mapping algorithm and heuristic algorithm

IP cores	RA-Map		DPSO			SA		
	T_{norm} (hop)	Time (s)	T_{norm} (hop)	Cost normalized to RL-MAP	Time (s)	T_{norm} (hop)	Cost normalized to RL-MAP	Time (s)
8	1.294	0.53	1.294	1.0	0.87	1.294	1.0	0.66
12	1.375	0.62	1.375	1.0	1.04	1.375	1.0	1.473
18	1.293	2.18	1.334	1.032	47.05	1.293	1.0	46.16
27	1.368	5.39	1.449	1.059	83.48	1.492	1.091	81.44
31	1.345	6.47	1.406	1.045	115.2	1.479	1.100	112.3
46	1.594	9.64	1.638	1.028	320.3	1.727	1.083	318.2
50	1.754	10.04	1.891	1.078	388	1.904	1.086	386.2
73	1.921	12.58	2.219	1.155	8897	2.321	1.208	8104
106	2.106	20.56	2.359	1.120	13073	2.396	1.138	11895
124	2.508	28.28	2.834	1.130	18436	2.883	1.150	16042
	Average			1.065			1.085	

From Table 5-2, it can be concluded that when the number of IP cores in the data acquisition system is less than or equal to 12, there is no difference between the mapping results of the RA-Map algorithm and the discrete particle swarm algorithm and simulated annealing algorithm. However, when the number of IPs is greater than 12, and when the number of IP cores is higher, the communication cost of the data acquisition system is smaller for the RA-Map mapping algorithm solution sequence compared with the mapping solution sequence of the discrete particle swarm algorithm and the simulated annealing algorithm. the average communication cost of the data acquisition system is reduced by 6.5% and 8.5% for the RA-Map mapping algorithm compared with the discrete particle swarm algorithm and the simulated annealing algorithm, respectively. 8.5%. In addition, in data acquisition task maps with a high number of IP cores, the RA-Map algorithm has a much shorter running time compared to other heuristics.

6. Conclusion

This article proposes a 3D NoC data acquisition system mapping algorithm based on reinforcement learning and improved attention mechanisms. Firstly, the algorithm improves the encoding method of nodes in the task graph. Secondly, the local attention mechanism is used in the mapping network encoder to overcome the problem of unnecessary information introduced by global attention mechanisms in the intermediate action vector. The mapping environment of the mapping network decoder is composed of global information, local information, and dynamic key node information of the task graph, to complete the selection of unmapped nodes. Finally, reinforcement learning is utilized to solve the challenge of obtaining high-quality training sets for the data acquisition system. Experimental results show that the RA-Map mapping algorithm reduces the average communication cost by 25.5% and 9.6%, respectively, compared to the encoder using global attention operations and the decoder not incorporating dynamic key node information into the mapping environment. Compared with the discrete particle swarm algorithm and simulated annealing algorithm, the communication cost of the RA-Map algorithm is reduced by 6.5% and 8.5%, respectively. The running time of the RA-Map algorithm is significantly reduced compared to other heuristic algorithms. Future work includes expanding the algorithm to different NoC topologies and implementing real-time communication energy, congestion, thermal sensing, and fault-tolerant application mapping.

Declarations

Ethical approval

We have not applied any tests on humans/animals in this research paper. So, no need for ethical approval.

Competing interests

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Authors' contributions

All authors contributed to the study conception and design. Funding acquisition,conceptualization and supervision were credited with by Chuanpei Xu. Material preparation, data collection and analysis were performed by Yang Wang. The first draft of the manuscript was written by Yang Wang, Xiuli Shi and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding

This work is supported by : National Natural Science Foundation of China (NSFC) (Grant numbers :62161008)

Availability of data and materials

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

[1] Choubey, Amit Kumar, and Ravi Raushan. "High-speed data acquisition system design. "2006 1st International Symposium on Systems and Control in Aerospace and Astronautics. IEEE, 2006.

[2] Zhao, J. , and C. Xu . "Design of time-interleaved data acquisition system based on Network on Chip." Concurrency and computation: practice and experience 10(2021):33.

[3] Amin, W. , et al. "Performance Evaluation of Application Mapping Approaches for Network-on-Chip Designs." IEEE Access PP.99(2020):1-1.

[4] Reza, Md Farhadur, et al. "Neuro-NoC: Energy optimization in heterogeneous many-core NoC using neural networks in dark silicon era." 2018 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2018.

[5] Ostler, Chris, and Karam S. Chatha. "An ILP formulation for system-level application mapping on network processor architectures." 2007 Design. Automation & Test in Europe Conference & Exhibition. IEEE, 2007.

[6] Ghosh, Pavel, Arunabha Sen, and Alexander Hall. "Energy efficient application mapping to NoC processing elements operating at multiple voltage levels ." 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip. IEEE, 2009.

[7] Huang, Jia , et al. "Energy-Aware Task Allocation for Network-on." Ifip International Conference on Advances in Production Management Systems Springer International Publishing, 2011.

[8] Tosun, Suleyman. "Cluster-based application mapping method for Network-on-Chip." Advances in Engineering Software 42.10(2011):868-874.

[9] Aravindhan, A., S. Salini, and G. Lakshminarayanan. "Cluster based application mapping strategy for 2D NoC. "Procedia Technology 25 (2016): 505-512.

[10] Lin, Ting-Jung, Shu-Yen Lin, and An-Yeu Wu. "Traffic-balanced IP mapping algorithm for 2D-mesh on-chip-networks." 2008 IEEE Workshop on Signal IEEE, 2008.

[11] Reshadi, Midia, Ahmad Khademzadeh, and Akram Reza. "Elixir: a new bandwidth-constrained mapping for networks-on-chip." IEICE Electronics Express 7.2 (2010): 73-79.

[12] Khan, Sarzamin, et al. "Bandwidth-constrained multi-objective segmented brute-force algorithm for efficient mapping of embedded applications on NoC architecture." IEEE Access 6 (2017): 11242-11254.

[13] Chatterjee, Navonil, Priyajit Mukherjee, and Santanu Chattopadhyay. "Reliability-aware application mapping onto mesh based Network-on-Chip." Integration 62 (2018): 92-113.

[14] Upadhyay, Mohit, et al. "Multi-application based network-on-chip design for mesh-of-tree topology using global mapping and reconfigurable architecture." 2019 32nd international conference on VLSI Design and 2019 18th international conference on embedded systems (VLSID). IEEE, 2019.

[15] Seidipiri, Rasoul, et al. "RASMAP: An efficient heuristic application mapping algorithm for network-on-chips." 2016 Eighth International Conference on Information and Knowledge Technology (IKT). IEEE, 2016.

[16] Sahu, Pradip Kumar, et al. "A constructive heuristic for application mapping onto mesh based network-on-chip." journal of Circuits, Systems and Computers 24.08 (2015): 1550126.

[17] Chen, Qingkun , et al. "An IP Core Mapping Algorithm Based on Neural Networks." IEEE Transactions on Very Large Scale Integration (VLSI) Systems PP.99(2020):1-14.

[18] Chen, Qingkun, et al. "A reinforcement learning-based framework for solving the ip mapping problem." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 29.9 (2021): 1638-1651.

[19] Jagadheesh, Samala, and P. Veda Bhanu. "Noc application mapping optimization using reinforcement learning. "ACM Transactions on Design Automation of Electronic Systems (TODAES) 27.6 (2022): 1-16.

[20] Sahu, Pradip Kumar, and Santanu Chattopadhyay. "A survey on application mapping strategies for network-on-chip design. "Journal of systems architecture 59.1 (2013): 60-76.

[21] Amin, Waqar, et al. "Performance evaluation of application mapping approaches for network-on-chip designs." IEEE Access 8 (2020): 63607-63631.

[22] Murali, Srinivasan, and Giovanni De Micheli. "Bandwidth-constrained mapping of cores onto NoC architectures." Proceedings design, automation and test in Europe conference and exhibition. vol. 2. IEEE, 2004.

[23] Jain, Arpit, et al. "Smart Communication Using 2D and 3D Mesh Network-on-Chip." Intelligent Automation & Soft Computing 34.3 (2022).

[24] Yang, Lei, et al. "Application mapping and scheduling for network-on-chip-based multiprocessor system-on-chip with fine-grain communication optimization." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24.10 (2016): 3027-3040.

[25] Williams, Ronald J. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Reinforcement learning (1992). 5-32.

[26] Vaswani, Ashish , et al. "Attention Is All You Need." arXiv (2017).

[27] Jiang, B. , et al. "Semi-Supervised Learning With Graph Learning-Convolutional Networks." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2020.

[28] Kingma, D. , and J. Ba . "Adam: A Method for Stochastic Optimization." Computer Science (2014).