Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/132/2024.20708

Adaptive Routing Algorithms of Network-on-Chip: A Literature Review

Yukun Sun[1,*], **Xueer Qian**[2], **Xintong Qu**[3]

[1]Beijing Academy, Beijing, China
[2]Fudan University, Shanghai, China
[3]Xidian University, Xi'an, China

[1]sophiayksun@outlook.com, [2]thspflaq@163.com, [3]1617872212@qq.com
[*]corresponding author

Abstract. To solve the challenges in the field of integrated circuits (IC) or system-on-chip (SoC) design, network-on-chip (NoC) has been recognized as a core concept. A routing algorithm is one of the key research among all the aspects NoC relates. Therefore, finding an optimal routing algorithm can help enhance the network's performance, reduce latency, increase scalability, decrease power consumption, limit expenditure, and so on. Due to its importance, we delve deeper into the area of routing algorithms for NoC designs. In this paper, we will review and summarize three mesh-based adaptive routing algorithms used for NoC, and provide a simple comparison and evaluation between these algorithms. Specifically, Neighbor-on-path (NoP) is used as the fundamental algorithm in our paper. Then the two advanced routing algorithms, regional ant colony optimization-based cascaded adaptive routing (RACO-CAR) and destination-based selection strategy (DBSS), will be analyzed and compared with NoP.

Keywords: Network-on-Chip, routing algorithms, adaptive routing

1. Introduction

With the ongoing advancement of semiconductor technology, the growing complexity of hardware and the delays caused by interconnections have become major challenges to the performance of System-on-Chip (SoC) designs. To address these challenges and enhance data transfer efficiency, the Network-on-Chip (NoC) architecture has emerged as a scalable, flexible, and reusable solution for chip multiprocessor (CMP) systems. NoC is a communication subsystem within an integrated circuit (IC) that enables data exchange between components like processors, memory units, and peripheral devices. As process technologies evolve and more cores are integrated into a single chip, the existing bus-based communication methods will no longer suffice. NoC is widely regarded as an optimal approach for developing modular and scalable communication architectures, with built-in support for integrating various core types by standardizing network interfaces [1].

Nowadays, more studies focus on the routing algorithm of NoCs due to the importance of finding the optimal one. Here are some of the reasons:

- **Enhancing Performance and Minimizing Latency:** Efficient routing algorithms are essential in minimizing the time for data packets to traverse the entire chip by selecting the shortest or fastest

DOI: 10.54254/2755-2721/132/2024.20708

available paths. By effectively distributing the load across various pathways, superior routing algorithms prevent bottlenecks and ensure optimal data throughput.

- **Energy efficiency:** Optimal routing reduces the number of hops and avoids congested paths, thereby minimizing power consumption. This is particularly critical for battery-operated and low-power devices. Furthermore, efficient routing can facilitate a more uniform distribution of power dissipation across the chip, aiding in thermal management.
- **Scalability:** As the number of cores or IP blocks increases, it is imperative for the routing algorithm to scale effectively to manage more intricate communication patterns without compromising performance. The implementation of scalable routing algorithms ensures that the NoC is capable of accommodating future enhancements and a higher core count.

As a result, finding an optimal routing algorithm has led to heated discussion and has become a research focus. Our paper compares different adaptive routing algorithms regarding several aspects. Our paper started with Neighbor-on-Path (NoP) selection, then compared this with two other routing algorithms which both made progress by overcoming the disadvantages of NoP selection. Regional ant colony optimization-based cascaded adaptive routing (RACO-CAR) and holistic routing algorithm, a destination-based selection (DBSS) strategy, are described in depth in the following passage. These two schemes will be published later than the NoP method, and all advanced NoP selections will be in various fields.

2. Background

2.1. NoC Architecture

The NoC technology applies computer networking theory and methods to on-chip communication, resulting in significant improvements over traditional bus communication architectures, and is available in various network topologies [1]. Most NoC architecture is now built on 2D-Mesh topology, as shown in Figure 1.

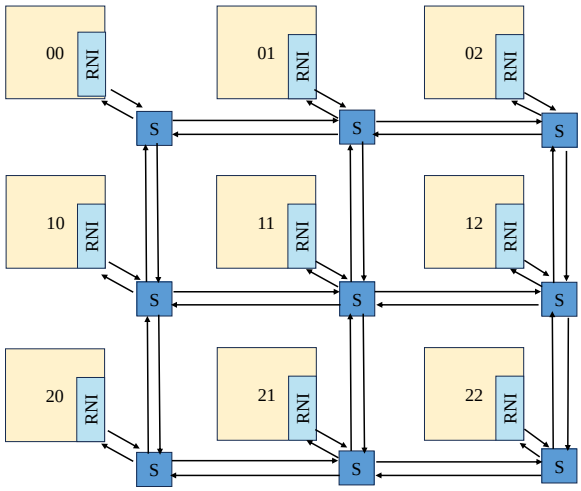


Figure 1: Structure of 3 x 3 2D-Mesh Topology

After reaching an intermediate node, a switching method helps to combine the input and output and form channels to set the switch. There are three widely recognized switching techniques: store-and-forward, virtual cut-through, and wormhole.

[1] Topology dictates the physical arrangement and interconnections among nodes and channels, and exerts a significant impact on both latency and power consumption. There are various topologies, such as mesh, ring, and star.

DOI: 10.54254/2755-2721/132/2024.20708

- **Store-and-Forward Switching:** In this method, the complete data frame is received and stored in a buffer before being sent to the destination. The switch verifies the frame for errors before forwarding it onward.
- **Virtual Cut-Through Switching:** In this method, the switch starts sending the data once the destination address are confined and enough of the message has been buffered to make a forwarding decision. However, there is a risk of forwarding corrupt or incomplete data if errors occur after the forwarding process has started.
- **Wormhole Switching:** Wormhole switching is a technique that combines aspects of both store-and-forward and cut-through switching. In wormhole switching, a small portion of the packet, known as a "worm," is initially stored and used to establish a path through the network. Once this path is set up, the data can start to flow through it before the entire packet has been received.

2.2. NoC Routing Algorithm

There are three main types of routing algorithms for NoC:

- Deterministic routing algorithm: These algorithms consistently give the same solution for a certain source and destination, such as XY routing and odd-even routing.
- Adaptive routing algorithm: These algorithms can choose different paths based on network conditions such as congestion or faults (west-first routing, minimal adaptive routing, etc.).
- Oblivious routing algorithm: These algorithms do not adapt to network conditions and follow predetermined paths (Valiant's randomized routing etc.).

While considering choosing an optimal routing algorithm, we usually focus on aspects, such as deadlock and livelock avoidance, fault tolerance, scalability, power efficiency, etc.

In our study, we concentrate on three types of adaptive routing algorithms specifically designed to dynamically adjust the path of data packets within a network based on current conditions such as congestion, faults, or traffic patterns. Through continuous monitoring of network conditions and real-time routing decisions, these algorithms play a crucial role in ensuring efficient, reliable, and fault-tolerant transmission.

An adaptive routing algorithm comprises two components as shown in Figure 2: the routing function and the selection strategy [2]. The current addresses and information are imported into the main routing function, as it calculates the number of potential output routes. Then, the selection strategy chooses one route throughout all the possibilities according to the network's current status to form a final output that turns out to be the most ideal [3].

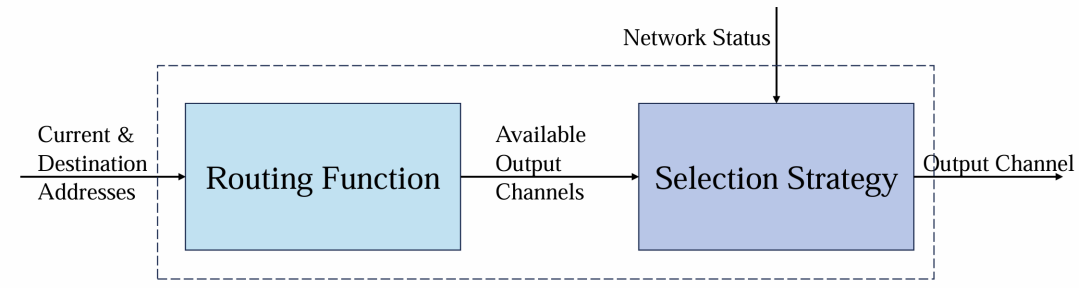


Figure 2: Structure of Adaptive Routing Algorithm

3. A New Method Relied on Adjacent Nodes in NoC

We review Neighbor-on-Path (NoP)[4] [2], combined with the Selection Function [5].

[2] an adaptive routing function considering Wormhole Switch on the mesh topology.

DOI: 10.54254/2755-2721/132/2024.20708

3.1. Analysis of NoP

N is the number of nodes in a network, *C* is the communication channels, and *PWC* represents the power set of C. The adaptive routing feature offers a group of alternate output channels. All flits within the data packet should be directed to these channels, as illustrated below:

$$N \times N \rightarrow PWC$$

To describe the NoP algorithm formally, they define some specific signals between two adjacent nodes and some parameters, referring to Figure 3. They are defined as follows:

- *free_slots_in[d]*: number of available free buffer slots for the adjacent node inputting in the direction d;
- *free_slots_out[d]*: number of available free buffer slots for current node inputting in the direction d;
- *NoPData_in[d]*: NoP-specific scores calculated by neighbor nodes in the direction d;
- *NoPData_out[d]*: NoP-specific scores for the current node by collecting input status information from adjacent nodes.

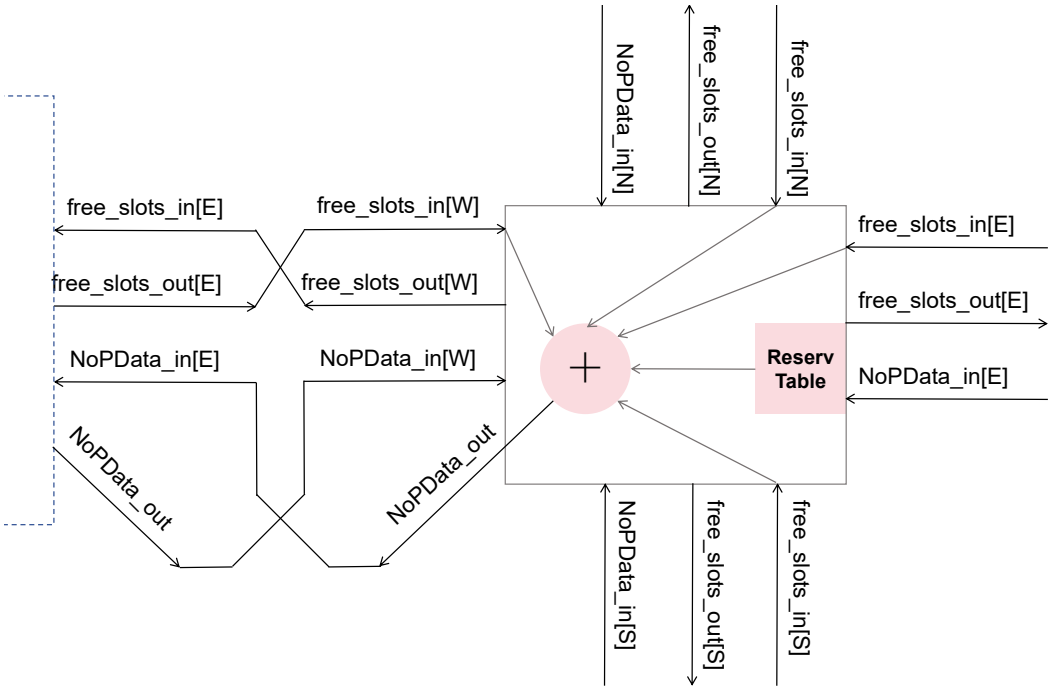


Figure 3: NoP selection strategy

Two pieces of the pseudocode of the Neighbor-on-Path which illustrates how the selection algorithm is implemented are shown below in algorithm 1 and algorithm 2. Please take note of the following text:

The algorithm collects information for adjacent neighbors from four directions, including *free_slots* and *available* (line 1). The value of *NoPData_out[d].free_slot* is obtained from *free_slots_in[d]*. It gives the number of accessible free buffer slots in the direction *d* (line 2). *NoPData_out.available[d]* is a Boolean function. If its value is ”0”, it represents the channel along the direction *d* is not reserved. On the contrary, if its value is ”1”, it represents the channel along the direction *d* is reserved (line 3). The local reservation table of the current router includes this value.

AOC is a set of available output channels that represents the input parameter. n_c is the current node and n_d is the destination node whose head flit must be routed. AOC is given by the function *R*:

$$AOC = R(n_c, n_d)$$

DOI: 10.54254/2755-2721/132/2024.20708

Each output channel (line 2) is checked by calculating how many NoP nodes are available (lines 3-4). After this, candidate channels are identified and confirm whether their input buffer is practical.

Algorithm 2 uses a scoring mechanism to evaluate the available channels at node n_c. *dest(ch)* is used to represent the target node starting from the channel *ch*. A scoring mechanism is implemented to select the candidate destinations with the highest score when there is space in the non-reserved input buffer (lines 6-8). If multiple channels are all maximum scores, then select randomly. That is how the best channel is selected. What should be mentioned is that there is no need to put data exchanges into global synchronizing. And what is glad to see is that data needs to be updated consistently and regularly.

Algorithm 1 ComputeNoPData

Require: $free\_slots\_in[]0$
Ensure: $NoPData\_out$
  for $d \in \{North, South, East, West\}$ **do**
    $NoPData\_out[d].free\_slots \leftarrow free\_slots\_in[d]$
    $NoPData\_out[d].available \leftarrow reservation\_table[d]$
  end for

Algorithm 2 NoP_Select

Require: AOC
Ensure: sc
  $scores[] \leftarrow 0$
  for $ch1 \in AOC$ **do**
    $node1 \leftarrow dest(ch1)$
    $AOC\_neighbor \leftarrow R(node1, nd)$
    for $ch2 \in AOC\_neighbor$ **do**
      if $NoPData\_in[ch1].available[ch2]$ **then**
        $score[ch1] += NoPData\_in[ch1].free\_slots[ch2]$
      end if
    end for
  end for
  $sc \leftarrow ch$ s.t. $score[ch] = max(scores[])$

3.2. Evaluation

To evaluate the performance of the NoP rooting algorithm, different traffic scenarios and routing methods are implied on a mesh-based NoC by using Noxim, an open-source SystemC simulator. In experiments both synthetic conditions and real traffic circumstances are involved. The synthetic conditions are the combination of uniform traffic, transposed traffic, and hotspot traffic, with their explanation as follows:

- The *uniform* traffic: each packets are sent equally.
- The *transposed* traffic: the packet on the node (*a*,*b*) only can be sent to the node (*T-1-a*,*T-a-b*). Here, a

  $$TtimesT$$

  mesh is implemented.
- The *hotspot* traffic: nodes which have been defined as the *hotspot* nodes received hotspot packets with *h* percent probability.

DOI: 10.54254/2755-2721/132/2024.20708

To evaluate the real communication traffic circumstance, a MultiMedia System (*mms*) is involved. *Packet injection rate*(*pir*) ranging from zero to one is the rate at which packets are injected. The *pir* is in the model of packets/cycle/node, and a *pir* represents that 0.1 packets are sent by every node in every clock cycle or a packet is sent by every node in every 10 clock cycles. Throughput and delay are used as performance metrics. Throughput is defined as follows:

$$TP = \frac{\textit{Total received flits}}{\textit{Number of nodes} \times \textit{Total cycles}},$$

where *Total received flits* is the sum of how many flits reach the targeted nodes, *The total cycle* is the sum of clock cycles that pass from the generation of the first message to the receive of the final message, and Total cycles is the number of clock cycles. *Delay* is the sum of clock cycles between the injection and the reception at the target node. Average delay, *D*, is shown as follows:

$$D = \frac{1}{K}\sum_{i=1}^{K} D_i,$$

where *K* is the sum of messages getting to their target nodes and D_i is the delay of message *i* [4]. 8-flit packets are injected into a 8×8 mesh NoC with random distribution. Every FIFO buffer has four flits. Before each beginning, the simulation is run for 10,000 cycles to keep stable and then executes for 20,000 cycles.

4. Regional ACO-Based Cascaded Adaptive Routing in Mesh-Based Network-on-Chip System

The algorithm introduced above (referring to NoP) didn't consider nor satisfy these two points while finding a better traffic-aware adaptive routing algorithm for NoCs: first, it didn't avoid the problem of using merely local network state information, which will result in unbalance, especially for those large-sized NoCs; second, it didn't avoid using only spatial network information while estimating congestion causing it not to be accepted in resource-constrained NoCs. However, the RACO-CAR algorithm, which will be introduced in the following paragraph, overcomes these two aforementioned points, and this method will then be elaborated closely.

4.1. ACO Introduction and Customization of Routing Algorithm

Ant colony optimization (ACO) is an algorithm that mimics the process of ants finding paths for food. When ants find food, they initially choose random paths. To optimize the foraging paths of ants, they perceive and are inclined to follow paths with higher pheromone concentrations, which the ant colony as a whole agrees upon as being superior. As shorter routes result in a shorter travel time for ants, more pheromones are released by more ants passing through these routes within a unit of time. Consequently, more ants will be inclined to follow these shorter route paths and leave their pheromones. Conversely, the pheromones on non-optimal routes will evaporate over time. By continuously updating their status, the entire ant colony will ultimately converge on the optimal path.

In the context of NoC routing, pheromone represents the probability of state transition when we make a routing decision. The pheromone itself is determined based on the traffic status of a local router, while the intensity of the pheromone can be described using a normalized probability distribution function. This is the state transition function between the new pheromone value and the old one:

$$P'(j,d) = \frac{P(j,d) + \alpha L_j}{1 + \alpha(|N_k| - 1)}, j \in \{north, east, south, west\} \quad (1)$$

where $P(j,d)$ is the pheromone value that determines the likelihood of sending a packet from the current router to a destination router d through channel j. L_j is the normalized value of output buffer

DOI: 10.54254/2755-2721/132/2024.20708

length at channel j, α is a weighting coefficient of historical and local network information, and N_k is the channel number in router k.

The RACO-CAR algorithm proposed in the paper [2] is based on the ACO selection function to choose a minimal path. Then, this routing algorithm was developed to minimize the occurrence of hotspots. In response to the challenges faced by traffic-aware adaptive routing algorithms, ACO not only utilizes pheromones to identify historically least-congested channels but also avoids the need for additional wires between routers for monitoring purposes. By simultaneously considering historical channel information and current buffer status in a distributed manner, ACO can predict traffic distribution and address congestion issues effectively, thereby integrating spatial and historical data as a congestion metric. Consequently, the use of an ACO-based algorithm can effectively achieve regional balance and overall efficiency.

4.2. Proposed Technique for Regional ACO

Compared to merely using the ACO algorithm, **regional ant colony optimization (RACO)** is being proposed and composed with two techniques, the **table elimination technique (TET)** and the **table sharing technique (TST)**.

While dealing with distant routers, the pheromones may become unreliable due to inaccurate predictions. Therefore, employing TET can eliminate certain pheromones for remote destinations in order to simplify the routing table, reduce memory usage, and minimize performance degeneration. Specifically, the original routing table is transformed from a square into a diamond shape when the diagonal distance of the region exceeds or equals the average hop count across all source-destination pairs in a mesh network. TET not only reduces the overall size of the table and search overhead but also ensures minimal performance degradation by making the distance of processing elements that frequently communicate with each other closer.

Conversely, when facing adjacent routers, similar pheromone behavior may occur; hence utilizing TST can merge neighboring packets with comparable conditions. This technique combines the NoC router into regions, and this is the compression ratio equation

$$TCR_{TST} = \frac{N_E}{k^2 - 1} \tag{2}$$

(k is the length of the mesh and N_E is the number of entries). Integrating the ACO algorithm with TET and TST gives the proposed algorithm RACO, it can achieve a smaller performance degradation and a lower overhead of searching tables than merely using only one of the techniques.

4.3. RACO-Based Cascaded Adaptive Routing

Then, the RACO algorithm is combined with the **cascaded adaptive routing (CAR)** method to compensate for the inaccuracy of distant destinations.

Firstly verify if the destination falls within the observation region. If not, the algorithm helps select a cascaded region from a pool of options and utilizes pheromone levels to designate an output channel for packet forwarding. Lastly, re-evaluate if the destination is within the region that we could observe. But if this doesn't work, the program will return to step 2, otherwise directly route the packet to its destination and conclude the process. This whole process can be abstracted into flow chart Figure 4:

The **minimum congestion searching (MCS)** algorithm is then applied to consider the traffic status. It serves to increase the adaptability for the search for uncongested cascaded regions, offset the performance loss caused by the two techniques as they eliminate some tables, and help to make a more precise routing decision

4.4. Implementation and Evaluation

The main pathway of the RACO-CAR architecture starts from the read pointer register of the FIFO (First In, First Out) storage and ends at the register that stores the grant signal within the switch allocator.

DOI: 10.54254/2755-2721/132/2024.20708

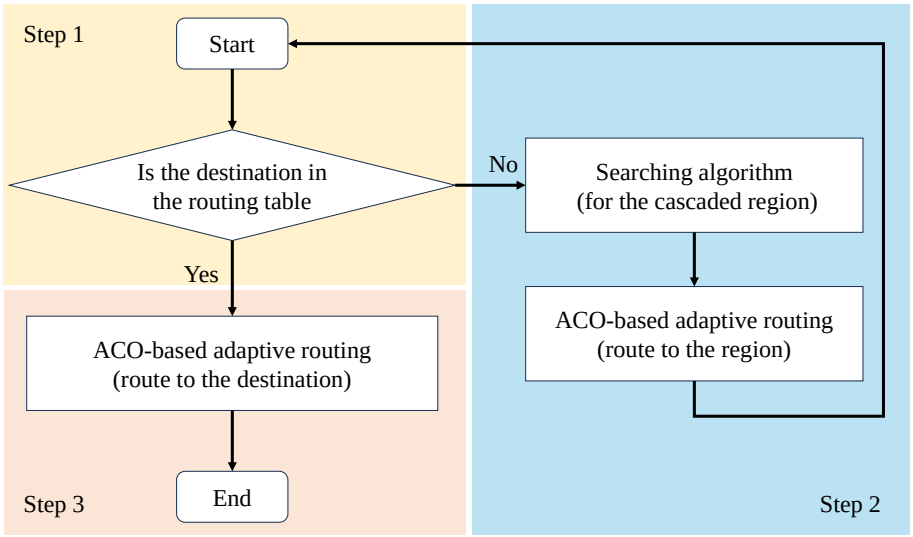


Figure 4: Flow Chart of Cascaded Adaptive Routing

The RACO-CAR algorithm is formed by integrating all the algorithms previously discussed. This study uses the Noxim Network-on-Chip (NoC) simulator to evaluate performance, comparing it against uniform random, transpose, and hotspot traffic patterns. Under *hs-row* traffic conditions, the NoP selection approach, unlike the RACO-CAR method, does not account for nodes constrained by the odd-even turn model. Instead, it prioritizes the output channel for packet forwarding, causing most packets to be sent through channels in the same row as the hotspot nodes. However, this preference can create congestion in certain channels along the row with these hotspot nodes.

Additionally, RACO-CAR exhibits a smaller latency than other routing algorithms. Furthermore, as the network size increases, the network throughput [3] of RACO-CAR steadily grows, indicating an improved scalability of performance and potential. Conversely, the NT of NoP and many other schemes gradually becomes precarious as network size increases.

However, the area overhead and power consumption levels using the RACO-CAR algorithm don't advance the NoP selection method very much, which are some potential aspects of future research and improvements. However, these two rates are still on a reasonable scale when applying RACO-CAR. In summary, the RACO-CAR algorithm reduced table cost significantly while achieving optimal tradeoffs between cost and performance.

5. A High Performance Selection Strategy with Low Power Consumption: DBSS

Several features are required in order to make a high-performance routing algorithm [3]: the algorithm should fully use the path diversity to avoid network congestion; it should not use unnecessary information causing inaccurate evaluation of network status; the algorithm could be implemented with low hardware overhead. Workload consolidation additionally requires another feature: dynamic isolation between different applications. However, most of the current algorithms are not able to satisfy these features.

5.1. Existing Problems

There already exist selection strategies like LOCAL or NoP, which only leverage the information of the neighboring nodes or the nodes adjacent to neighboring nodes. The main disadvantage of these selection strategies is that the performance will degrade due to limited information with a fully adaptive routing

[3] *network throughput (NT) = total received flits / total cycles*

DOI: 10.54254/2755-2721/132/2024.20708

function. They particularly perform worse with a smaller Average Hop Count (AHP)[4]. There are also strategies like Regional Congestion Awareness (RCA), which use global information to decide the path. Although this kind of strategy performs better with a larger AHP, it introduces interference both within and between regions, which degrades the performance and the predictability of performance.

5.2. DBSS

Therefore, a new selection strategy, Destination-based Selection Strategy (DBSS), is proposed in the paper [3]. It is given high adaptability with a low-cost congestion information propagation network, and it dynamically isolates different applications by ignoring nodes that are not in the smallest rectangular area defined by the current location and the destination.

Figure 5 shows the examples of 4 selection strategies, LOCAL, NoP, RCA-1D[5] and DBSS in a 4×4 mesh when the current location is (0, 2) and the destination is (2, 0). When we need to choose from the west and the south ports, the shadowy areas indicate the nodes whose information is taken into account by each selection strategy. LOCAL only considers the nearest nodes, (0, 1) for the west port evaluation and (1, 2) for the south port evaluation. NoP considers the second nearest nodes, (0, 0) and (1, 1) for the west and (1, 1) and (2, 2) for the south. This makes them unable to avoid congestion more than 1 or 2 hops away. RCA, on the contrary, considers all the nodes in one direction, (0, 1) and (0, 0) for the west and (1, 2), (2, 2) and (3, 2) for the south. This makes RCA have good performance with a larger AHP when the global information matters more. However, RCA introduces interference as it uses an excess amount of information. In this case, node (3, 2) is taken into account although it will probably not be traversed by the packet and intraregion interference is introduced. Even, if multiple applications are working on different regions, RCA considers nodes in other regions as Figure 6 shows. This interregion interference has a negative influence on dynamic isolation. Also, both intraregion and interregion interference degrades the performance.

DBSS considers nodes inside the minimum quadrant, (0, 1) and (0, 0) for the west and (1, 2) and (2, 2) for the south. It uses some global information and mitigates interference simultaneously, which makes it perform well generally with different AHPs.

5.3. Routing Functions

The performance and efficiency of a selection strategy are tightly connected with its underlying routing function, as the routing function provides the available paths to choose from and controls VC reallocation. Therefore, when discussing selection strategies, we must look at routing functions as well.

Adaptive routing functions can be classified into 2 kinds, partially adaptive ones and fully adaptive ones. For partially adaptive routing functions, deadlocks are avoided by prohibiting some turns. For example, negative-first, west-first, north-last, and odd-even (which means different turns are forbidden in odd and even columns) are introduced. On the other hand, fully adaptive routing functions use a different way to avoid deadlocks: they restrict VCs. According to Duato's theory, VCs are classified as adaptive and escape VCs. While no limitations are placed on adaptive VCs, escape VCs will only be used in deadlock-free algorithms.

5.4. Evaluation

In the paper, BookSim is used to simulate the algorithms. The evaluation comprises 2 parts as follows.

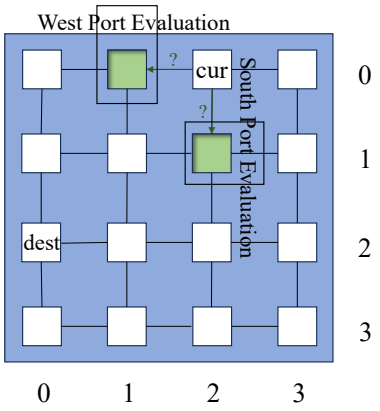
(i) Evaluation of different routing functions

(ii) Evaluation of different selection strategies

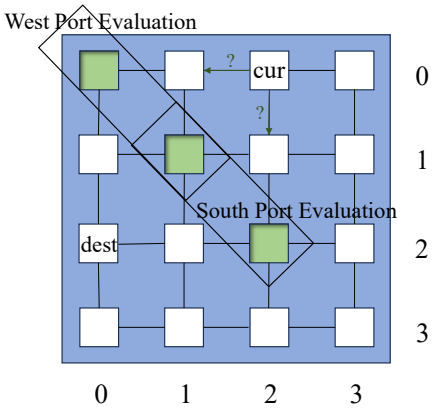
[4] AHP can be seen as an indicator of how long on average a packet travels from the start point to the destination.

[5] There are many RCA variants. In this paper, we mainly talk about RCA-1D. RCA-1D will be abbreviated as RCA hereafter.

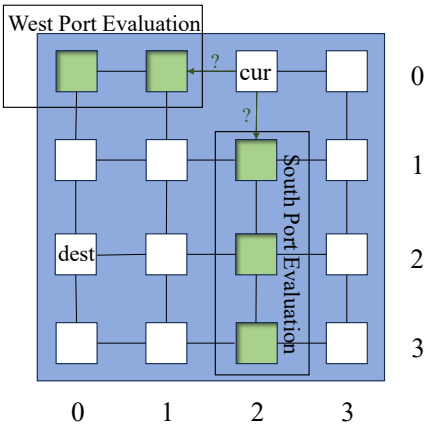
DOI: 10.54254/2755-2721/132/2024.20708



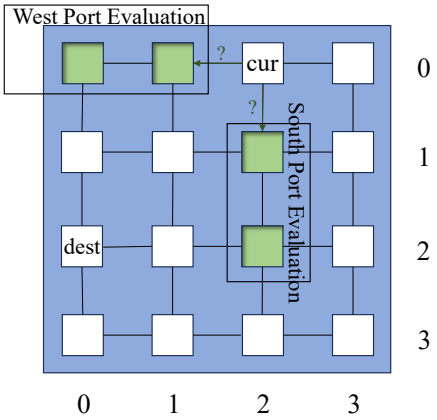
(a) A Situation of LOCAL.



(b) A Situation of NoP.



(c) An Intraregion Interference Situation of RCA-1D.



(d) A Situation of DBSS.

Figure 5: Examples of Several Selection Strategies.

5.4.1. Evaluation of Different Routing Functions Negative-first, west-first, north-last, and a fully adaptive function based on Duato's theory (abbreviated as Duato hereafter) are evaluated with a LOCAL selection strategy.

The evaluation result shows that Duato performs better with the increase in VC numbers. For the 2-VC situation, partially adaptive functions generally perform better than Duato because they make full use of VCs while Duato saves half of the VCs to be only used under deadlock-free algorithms. For 6 or 8 VC situations, however, Duato outperforms partially adaptive functions since VC numbers are no longer the bottleneck. Instead, the path diversity of the fully adaptive function causes higher performance.

DOI: 10.54254/2755-2721/132/2024.20708

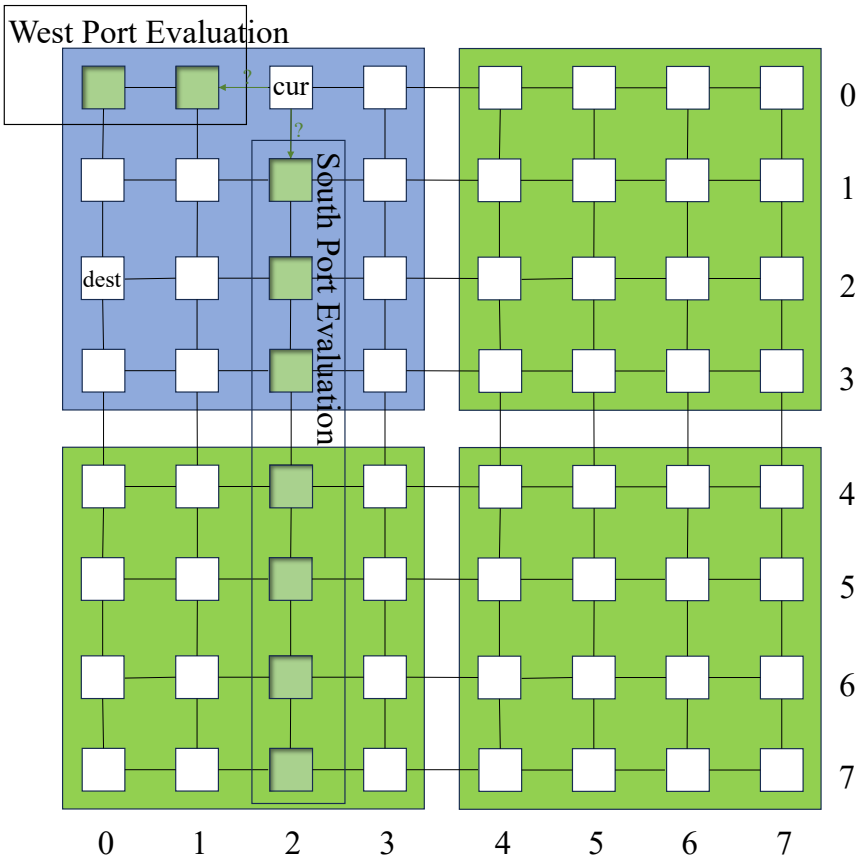


Figure 6: An Interregion Interference Situation of RCA-1D.

5.4.2. Evaluation on Different Selection Strategies From step 1, we know that Duato with 6 or 8 VCs performs the best among all situations simulated. Thus, in this part of the evaluation, we select Duato as the routing function with 8 VCs.

The result shows that in transpose-1, bit reverse, and shuffle traffic, DBSS all perform the best. Nevertheless, RCA performs better than DBSS in bit complement traffic. This is related to the large AHP of bit complement compared to other kinds of traffic. As mentioned before, RCA tends to perform better with larger AHP when global information matters more. This can also be seen from the fact that in the 4×4 mesh, LOCAL and NoP outperform RCA except for the bit complement, while in the 8×8 mesh, RCA outperforms LOCAL and NoP.

On the contrary, LOCAL and NoP selection strategies tend to perform better with smaller AHP, because they focus on nearer nodes and ignore global information. When the AHP is smaller, the

DOI: 10.54254/2755-2721/132/2024.20708

knowledge of nodes far away does not matter much, and the information of nearer nodes takes the leading role instead.

But generally, DBSS shows the best performance in most cases, as it combines the advantages of LOCAL/NoP and RCA, being able to consider global information while avoiding interference simultaneously.

Besides, multiple-region situations are also simulated. LOCAL, NoP, and DBSS all perform the same as in single-region configuration because they have no interregion interference. RCA, however, performs worse than in single-region configuration as it introduces interregion interference.

5.5. Hardware Overhead

The hardware overhead is a significant aspect in assessing a selection strategy because it directly influences the cost to implement the selection strategy in practice.

5.5.1. Wiring Overhead As adaptive routing algorithms need congestion information to decide the path, some wiring overhead is needed to transmit the information. Considering an 8×8 mesh with 8 VCs, the wiring overhead of different selection strategies are as follows. We will look at 2 major parts of hardware overhead: the wiring overhead and power consumption.

Table 1: Wiring Overhead of Different Selection Strategies

Selection Strategy	DBSS	RCA	NoP	LOCAL
Wire per Dimension	8	16	24	4

DBSS has a moderate wiring overhead much lower than RCA and NoP, and a little higher than LOCAL.

5.5.2. Power Consumption A 32 nm process and 1 GHz clock frequency are used in the evaluation of power consumption.

The evaluation result shows that the power used by DBSS is relatively low, lower than NoP and RCA, and similar to LOCAL. However, when we look at the energy-delay product (EDP), DBSS becomes the best. EDP is a metric used to assess the overall efficiency of a system. It is calculated by multiplying the energy consumption and the latency. Low EDP means that DBSS has a good balance of power consumption and performance.

Generally speaking, DBSS is a selection strategy with good performance, moderate hardware overhead, and low power consumption.

6. Conclusion

This paper reviews three representative adaptive routing algorithms: NoP, RACO-CAR, and DBSS all on mesh topology. We compare the performance and overhead of these three algorithms, and it can be concluded that both RACO-CAR and DBSS have a much better performance, even though they may cause some overheads. The selection of the routing algorithm is a balanced choice of performance indicators such as area cost, power consumption, throughput, latency, etc. This comparison can provide ideas and inspiration for researchers' future research directions to some extent.

Acknowledgment

Yukun Sun, Xueer Qian, and Xintong Qu contributed equally to this work and should be considered co-first authors.

The authors would like to thank Professor William Nace for his clear and understandable teaching, kindness, and patience in answering our questions. The authors would also like to thank TA Shang Liu and Xiaoyang Wang for their kind help and advice in many aspects.

DOI: 10.54254/2755-2721/132/2024.20708

References

[1] P. Guerrier and A. Greiner. A generic architecture for on-chip packet-switched interconnections. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000 (Cat. No. PR00537)*, pages 250–256, 2000.

[2] En-Jui Chang, Hsien-Kai Hsin, Chih-Hao Chao, Shu-Yen Lin, and An-Yeu Wu. Regional aco-based cascaded adaptive routing for traffic balancing in mesh-based network-on-chip systems. *IEEE Transactions on Computers*, 64(3):868–875, 2015.

[3] Sheng Ma, Natalie Enright Jerger, Zhiying Wang, Mingche Lai, and Libo Huang. Holistic routing algorithm design to support workload consolidation in nocs. *IEEE Transactions on Computers*, 63(3):529–542, 2012.

[4] Giuseppe Ascia, Vincenzo Catania, Maurizio Palesi, and Davide Patti. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *IEEE transactions on computers*, 57(6):809–820, 2008.

[5] Maurizio Palesi, Rickard Holsmark, Shashi Kumar, and Vincenzo Catania. Application specific routing algorithms for networks on chip. *IEEE Transactions on Parallel and Distributed Systems*, 20(3):316–330, 2009.