Front. Comput. Sci.
DOI 10.1007/s11704-016-5431-8

REVIEW ARTICLE

A survey of routing algorithm for mesh Network-on-Chip

Yue WU[1,2], Chao LU[2,3], Yunji CHEN (✉)[1]

1 State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
2 School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408, China
3 Loongson Technology Corporation Limited, Beijing 100095, China

Abstract With the rapid development of semiconductor industry, the number of cores integrated on chip increases quickly, which brings tough challenges such as bandwidth, scalability and power into on-chip interconnection. Under such background, Network-on-Chip (NoC) is proposed and gradually replacing the traditional on-chip interconnections such as sharing bus and crossbar. For the convenience of physical layout, mesh is the most used topology in NoC design. Routing algorithm, which decides the paths of packets, has significant impact on the latency and throughput of network. Thus routing algorithm plays a vital role in a well-performed network. This study mainly focuses on the routing algorithms of mesh NoC. By whether taking network information into consideration in routing decision, routing algorithms of NoC can be roughly classified into oblivious routing and adaptive routing. Oblivious routing costs less without adaptiveness while adaptive routing is on the contrary. To combine the advantages of oblivious and adaptive routing algorithm, half-adaptive algorithms were proposed. In this paper, the concepts, taxonomy and features of routing algorithms of NoC are introduced. Then the importance of routing algorithms in mesh NoC is highlighted, and representative routing algorithms with respective features are reviewed and summarized. Finally, we try to shed light upon the future work of NoC routing algorithms.

Keywords Network-on-Chip, mesh topology, routing algorithm, adaptive routing, oblivious routing

Received October 15, 2015; accepted February 17, 2016

E-mail: cyj@ict.ac.cn

1 Introduction

Benefiting from the continuous improvement of processing technology in semiconductor industry, increasing number of cores are integrated on chip, many traditional challenges as power and bandwidth rise up again, thus on-chip interconnection becomes the critical aspect of chip performance. Profound changes are taking place in on-chip interconnection. Therefore, NoC, proposed as a resource management solution with high scalability, is gradually replacing the traditional on-chip interconnection methods such as bus and crossbar. Various contributions have been made in this area in recent years [1–6].

Different from traditional on-chip interconnection, NoC implements packet-switch communications as distributed network. Data are divided into packages and routed from the source node to the destination node by routers. The corresponding research areas of NoC, i.e., system, network adapter, network and link, are listed in Table 1 [7]. The corresponding open system interconnection (OSI) protocol levels are also appended.

For the simplicity of physical layout and short wire lengths, mesh topology is most widely adopted in actual implementations. In this paper, we focus on mesh topology and its derived ones, e.g., torus and concentrated mesh. Once the topology of a NoC is fixed, the main concerns of the design are routing algorithm and flow control mechanism.

Routing algorithm defines the strategy of transferring data packets from source node to destination node. Routing algorithm has great influence on network efficiency, workload balance and error robustness, playing an important role in the whole system. A well designed routing algorithm should have the properties with reasonable routing path lengths, tolerable worst packets latency, low average latency, and good load balance in various types of workload to maintain high network throughput, well-tuned with flow control mechanism to avoid and resolve network transfer errors, deadlock especially. Therefore, routing algorithm has become a research hotspot ever since NoC was proposed. There are plenty of research conclusions in various aspects. Though routing algorithms of NoC and the traditional distributed networks share some background properties, there still remains environmental difference essentially. Distributed networks often have larger scale and irregular topologies with delay tolerance, while NoC tends to apply smaller scale and more regular topology with less delay tolerance.

Table 1 Research area of NoC

Research level	Research area	OSI protocol level
System	Design methodology Architecture domain Traffic characterization	Application Presentation
Network adapter	Functionality Sockets	Session Transport
Network	Topology Protocol Flow control QoS	Network
Link	Synchronization Reliability Encoding Data	Data

So far as we know, no one has specialized in reviewing the routing algorithm of NoC in detail. In order to face the challenge brought by the increasing complexity of processor design, a complementary study on routing algorithms of mesh NoC is of great significance, which will be discussed in the following.

Most of the earliest routing algorithms on NoC, e.g., dimension-order routing (DOR) [8] and zigzag [9], belong to oblivious routing. They are relatively simple and of negligible cost to implement. However, the algorithm works poorly with unbalanced network workload since the routing path is a fixed function of the source node and destination node. To alleviate such negative effect, random factors are introduced into oblivious algorithms such as ROMM [10].

All these algorithms above cannot dynamically balance the network's workload. To change this, the adaptive routing algorithms are proposed. Adaptive routing algorithms select paths by monitoring realtime workload. The early version, LOCAL [11,12] for instance, only adopts the congestion information of nodes adjacent to current node. To issue this defect, some research has proposed algorithms such as RCA [13], DBAR [14] which leverage more network information, even global information. Kakoulli et al. [15] proposed an self-adaptive algorithm which leverages global network information proceeded by artificial neural network (ANN).

Compared with oblivious routing, adaptive routing algorithms adapt to a more varied network environment and achieve better performance when the network's workload is high or unbalanced. Meanwhile, adaptive routing has to pay the price for the extra logic overhead in acquiring information, path arbitration and deadlock avoidance. As a tradeoff between adaptive algorithms and oblivious algorithms, the half-adaptive routing algorithms are proposed. Half-adaptive algorithms could switch from one oblivious algorithm mode to another, depending on the evaluation of current workload. Half-adaptive algorithms make significant reduction of complexity and overhead, though lose some path diversity.

The paper is organised as follows. Section 2 introduces the basic concepts and related technical features of mesh NoC structure. Sections 3–5 give descriptions of various research branches on NoC routing algorithms. Section 6 tries to shed light on the future research trend of NoC routing algorithms.

2 General view of routing algorithms for mesh NoC

2.1 Concepts and taxonomy of NoC routing algorithms

Routing algorithm can be formalized as a routing relation R and a selection function ρ [16]. The relation R is defined as a mapping:

$$N \times N \to \mathcal{P}(C), \tag{1}$$

where N donates the set of network nodes, C represents the set of all available channels and $\mathcal{P}(C)$ is the power set of C. Routing relation R returns a subset of C while selection function ρ selects a channel from the returned set by R, avoiding both deadlock and channel dependency. Figure 1 illustrates the relationship between routing relation and selection function.

Figure 2 shows the taxonomy of routing algorithms. Routing algorithms of NoC can be generally divided into two main categories, i.e., oblivious algorithms and adaptive algorithms, depending on whether the current network status affects path selection. Oblivious algorithms select the routing path in a stationary way, while adaptive algorithms takes current workload of network into consideration when routing. Deterministic algorithms is a subset of oblivious ones, in which the package path selection can be regarded as a function of the source nodes and destination nodes. The chosen path is preordained once the source node and destination node are determined. For example, dimensional ordered routing (DOR) is a typical oblivious routing algorithm [10]. In adaptive routing algorithms, the chosen package paths varies by the current network status. We divide adaptive routing algorithms into two subtypes, global adaptive routing and local adaptive routing based on the scale in which the network status is fetched. Though oblivious routing is widely applied in practical, researchers pay much attention to adaptive routing, and in this paper we mainly focus on adaptive routing algorithms.

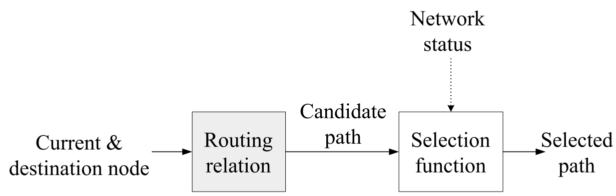


Fig. 1 Structure of NoC routing algorithms

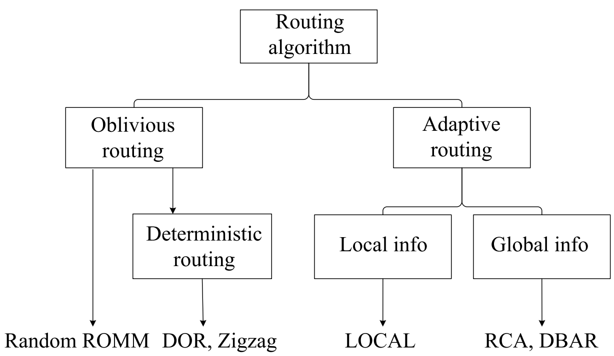


Fig. 2 Taxonomy of routing algorithms

According to the lengths of paths selected by function ρ, routing algorithms can be labelled as minimal routing or non-minimal routing. Minimal routing algorithm always selects paths with minimum length, while non-minimal routing may choose a longer path. Though showing great advantage in path diversity, non-minimal routing has much more complex mechanisms to achieve deadlock avoidance, which limits non-minimal routing's implementation. In this paper, we mainly discuss minimal routing algorithms.

Specifically, in 2D mesh topology, the routing relation of minimal routing only returns two of the possible ports/directions at most, and the selection function chooses one from the two, as illustrated in Fig. 3. In this circumstance, it is clear that selection function is key to the performance algorithm.

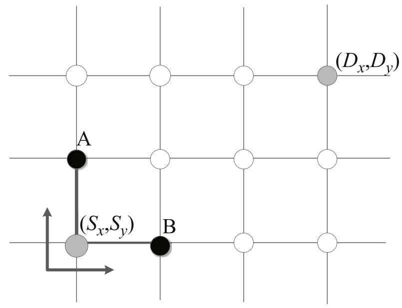


Fig. 3 Selection function of minimal routing algorithm in 2D mesh

For oblivious routing algorithms [8,9,17,18], selection functions are defined by a static selection rule S which is not concerned with the current network status. For instance, DOR [8], i.e., XY-routing/YX-routing, always transfers packets along the X/Y dimension with higher priority. Some algorithms randomly generate the direction of the next flip under certain constraints.

The selection function of adaptive routing algorithms varies with the networks' status. Simple adaptive routing algorithms [11,12] choose candidate output ports by monitoring congestion status of neighboring nodes. To achieve global-optimal and precise routing selection, sophisticated adaptive routing algorithms acquire information of non-local nodes to choose the output port [12–14,19,20].

Livelock and deadlock are side effects brought with more path diversities in adaptive routing algorithms. Livelock refers to the circumstance that routing path turns to an end-to-end loop, which means packets routed are impossible to reach their destinations[1]. Deadlock refers to the phenomenon that a loop exists in the hold-wait-for relation of the agent or network resources and unable to resolve the situation [16]. The most common mechanisms to avoid deadlocks include Turn Model proposed by Glass et al. [21] and the Virtual Channel Theory proposed by Duato et al. [22–24], which will be discussed in detail in Section 4.

In short, adaptive routing algorithm achieves better performance in various network workload with the cost of network monitoring, deadlock avoidance and routing arbitration. To combine the advantages of oblivious and adaptive algorithms, half-adaptive routing algorithms [25–27] are proposed. Some half-adaptive routing algorithms can be described as a selection in two or more oblivious routing models, which is per-

[1] It is crystal clear that minimal routing algorithms are livelock-free, which avoids the overhead of livelock avoidance/detection mechanisms

formed at the source node.

A well-designed routing algorithm is a synthesis of performance, complexity and robustness. Both oblivious routing and adaptive routing have its own advantages and disadvantages. For instance, DOR [8] and XY-routing need O(1) times routing selection, while DBAR [14] and greedy need O(n) times (n is the size of mesh network). The space complexities of acquiring congestion information in DOR and greedy are O(1) while those in XY-routing and DBAR are O(n). Table 2 compares the three major types of routing algorithms mentioned above.

Table 2 Comparison of routing algorithms

Routing algorithms	Oblivious routing	Adaptive routing	Half-adaptive routing
Complexity	Low	High	Median
Adaptability	Low	High	Median
Deadlock-free mechanism	Simple	Complex	Median
Throughput	Low	High	High

2.2 Metric for the performance of routing algorithms

Generally, balanced workload is suitable for oblivious routing algorithms, while the robustness of adaptive routing is best shown on imbalanced workload. We usually evaluate the performance network with metrics such as packet latency, and throughput. Synthesis input and benchmarks are both often injected into network in evaluation.

The synthesis modes are summarized and abstracted from application behavior. Common synthesis modes include *uniform*, *shuffle*, *transpose*, *neighbor* and so on. Synthesis input works in the following way:

1) Randomly choose the injected nodes (source node), and artificially generate a packet.
2) To generate the destination node d depending on the source node s, pre-defined injection mode and random factors.

Uniform mode distributes the workload to the whole network uniformly. However, in real system, running applications behaves with more outburst, unbalance and jitteriness [7,28–30]. Uniform mode balance the workload naturally and is not stressed enough compared to real applications. Under uniform mode, most routing algorithms performs well, and merits and demerits are hardly to tell [21,25,31,32]. Ununiform modes such as shuffle and transpose offer more unbalance to network, showing more difference between routing algorithms. The inject rate of these synthesis mode can be precisely controlled to test the algorithms in different network stress conveniently.

To achieve a more convinced result, benchmarks such as Splash-2 [33] or SPEC [34] are often simulated in NoC. The behavior of different applications varies in a wide range. For example, some computing bound applications in benchmarks generate little traffic in the network, leading to the insensitiveness to difference routing algorithms. On the contrary, some memory bound applications behave in a distinct way such as *fft*, *water-ns*, *water* and *lu* in Splash-2 [35].

3 General view of oblivious routing algorithms

For the simplicity and deadlock-free, DOR [8] is one of the most widely applied oblivious routing algorithms in distributed systems and NoCs. For each of the source-destination node pair, DOR always transport the packages along dimension with predefined priority, for instance, the x-axis direction in XY-routing and y-axis direction in YX-routing. The lack of path diversity sometimes performs poorly under unbalanced workload. Introducing random variables to enhance path diversity helps to weaken this defection.

Nesson et al. proposed ROMM [10] as an enhancement version of DOR, introducing random intermediate nodes between the source-destination node pair and path relation by introducing factors. In k-phase ROMM, the whole path (from source to destination) is split into k segments. In each segment, DOR is performed. Larger k makes ROMM more like the distributed routing and consumes more resources. Other enhanced algorithms based on DOR include VAL, RLB and LEF [17,18,36].

Zigzag [9] is another commonly used oblivious routing algorithm. In each intermediate node, the algorithm keeps on choosing the candidate nodes with the longest dimensional distance, which implies that Zigzag would conduct the most turns. Some of the oblivious algorithms are illustrated in Fig. 4.

4 General view of adaptive routing algorithms

4.1 Local adaptive routing algorithm

The advantage of adaptive routing lies in the awareness of network status. To evaluate the congestion of different nodes, an appropriate and practical metric is in dire need. Different metrics including free buffer, free virtual channel, crossbar request and their combinations are applied in previous research [12–14,37,38]. Gratz et al. [13] stated that with a single metric, free buffer or free virtual channel performs equally and crossbar request works slightly better. An appropriate combination of multiple metrics would outperform any single one.

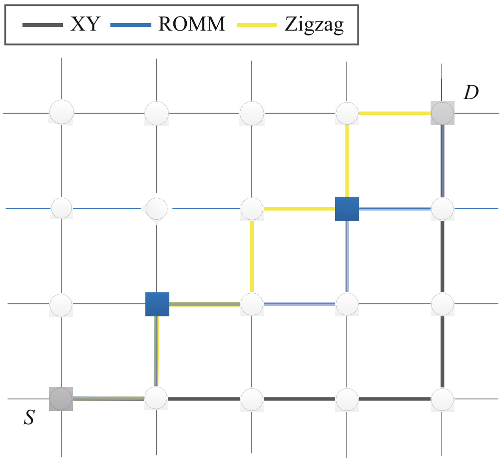


Fig. 4 Dimension-order routing, 3-phase ROMM and Zigzag

LOCAL [11,12] is simple greedy algorithm with workload balance. In each hop, by comparing the congestion status of the two candidate node, the algorithm always chooses the less congested direction. For example, as shown in Fig. 5, the source node is (0,0) and destination (2,3). The data are transferred to (1,0) because of its less congestion along the x-axis at the first hop. Similarly, the following hops are (1,1) and (2,1). Later on, there is only one possible direction, so the last two hops are (2,2) and (2,3).

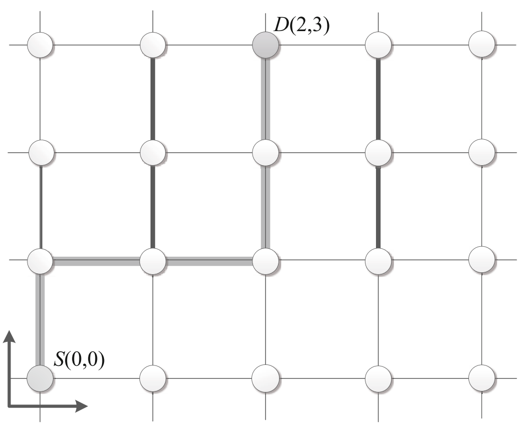


Fig. 5 LOCAL routing algorithm

4.2 Global adaptive routing algorithm

LOCAL is not global optimal since it might choose a path with low congestion near the current node but high congestion along the further part of the path. As shown in Fig. 5, LOCAL algorithm performs a non-global optimal arbitration at node (0,1). To address this problem, global adaptive routing algorithms are proposed.

Instead of only monitoring the current node and the adjacent nodes, global adaptive routing acquires congestion status of neighboring area in a larger scale, even in global scale [10,13,14,19,20]. Global adaptive routing gives consideration to not only the current hop but also the several future hops followed. Global adaptive routing is an improvement of the simple and direct greedy algorithms such as LOCAL, and outperforms better in network with large or unbalanced workload.

RCA proposed by Gratz et al. [13] is one of the global adaptive routing algorithms. As shown in Fig. 6, the congestion status flow is the opposite direction of data flow. Each node gets its own congestion metric, calculates a weighted average of local congestion metric and contention values from adjacent nodes, and propagates the weighted average value to the upstream node.

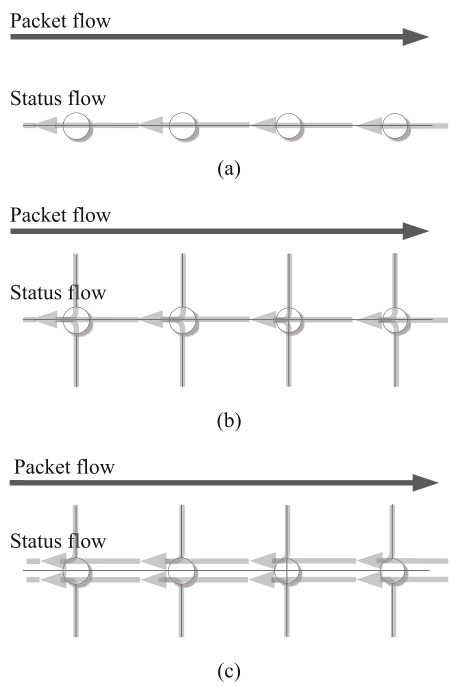


Fig. 6 Three different weight values for aggregation of RCA. (a) 1D; (b) Fanin; (c) Quad-rant

According to the different weight values for aggregation, there are three kind of RCA, which are 1D, Fanin and Quadrant, as shown in Fig. 6. For example, RCA-1D with local weight 0.5, $R_{i+1} = 0.5 \times R_i + C_{i+1}$, C_i is the congestion metric of the ith node along the congestion status flow.

One of the disadvantages of RCA is that the contention values contain some unnecessary information since the calculating process brings in the congestion status information of irrelevant nodes. RCA has to add plenty of status links for the propagation of congestion information.

Gratz et al. [2] proposed Neighbors on Path (NoP) as another solution, which selects ports one hop ahead of LOCAL. NoP algorithm monitors the congestion of the adjacent nodes two hops away. For example, as Fig. 7(a) illustrates, the current and destination node are is (1,1) and (2,3), and the candidate nodes are (1,2) and (2,1) respectively. NoP evaluates the congestion information of node pairs (1,3)(2,2) and (2,2)(3,1) separately, in the view of node (1,2) and node (2,1).

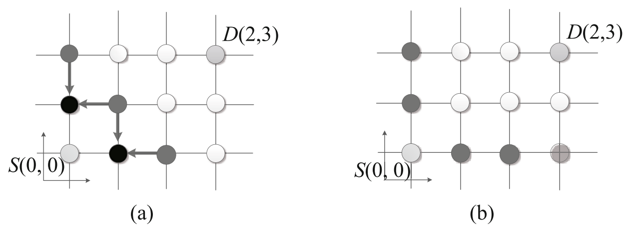


Fig. 7 Comparison of (a) NoP and (b) DBAR algorithms in areas of global information

Ma et al. proposed DBAR [14] as an improvement based on RCA for multi districts in NoC. DBAR focus on the congestion status of nodes in the same row/column in order to avoid redundant information, as shown in Fig. 7(b). DBAR stores congestion information of the row and column in two group of registers, $Congest_X$ and $Congest_Y$, one bit for one node. Assuming the source node is (x_s, y_s) and destination is (x_d, y_d), and $y_d > y_s$, $x_d > x_s$, then the congestion values of the two directions are derived: $Cong_x = Congest_X[x_s : x_d]$ and $Cong_y = Congest_Y[y_s : y_d]$. Zeros will be added to the tail of the shorter one between $Cong_x$ and $Cong_y$, keeping $Cong_x$ and $Cong_y$ the same length. CATRA [39] proposed by Ebrahimi et al. is an intriguing comparison with DBAR, which uses both local and non-local congestion information in a more updated way.

All of these algorithms mentioned above collect and propagate congestion status of related nodes. Similar algorithms include those proposed by [12,19,40,41]. To physically implement these routing algorithms, extra modules for monitoring, aggregation and propagation are implemented in the form of accompanied network, global routing table [16,42,43], point-to-point communication, etc.

Both RCA and DBAR leverage embedded monitoring networks. To achieve the management of accompanied network, aggregation and propagation modules are implemented in micro architecture of routers. Figure 8 shows the micro architecture of conventional router and router implemented in these algorithms. As shown in Fig. 8(b), aggregation module of RCA receives congestion status information of the downstream routers. Congestion status downstream and local virtual channel occupation status are inputs of the calculation logic. The calculation logic outputs the congestion estimation of each port to pre-selection logic and propagation module. Propagation module adds the weighted results of aggregation module, and outputs the results to the aggregation modules downstream. DBAR uses two groups of register integrated in the router, Congest X and Congest Y, to store the congestion status of the row/column the node belongs to. The two groups of registers from adjacent nodes are connected. Meanwhile, the selection algorithm uses these registers as input and calculates the out-dim vector, which is similar to the routing table in some way, and send the vector to the preselection logic, as

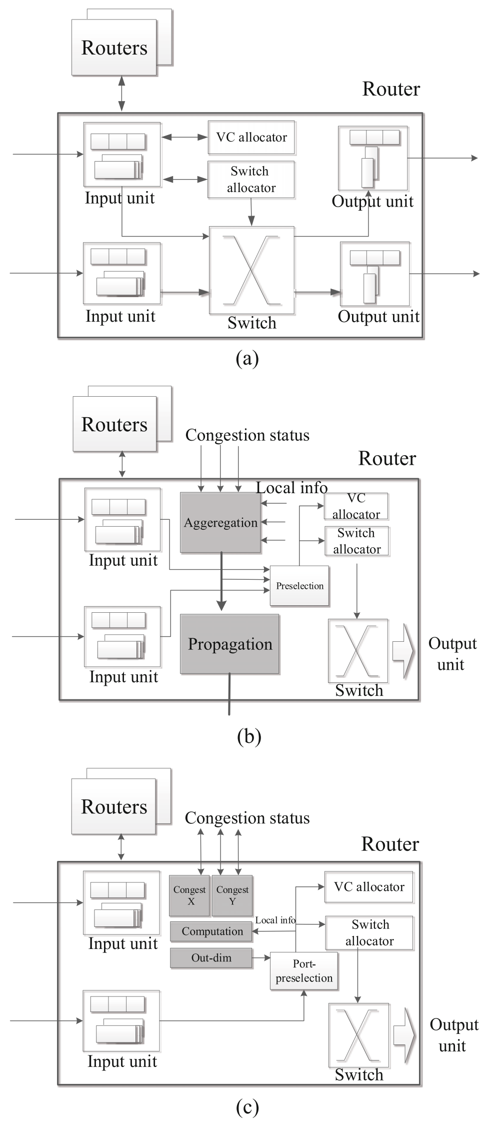


Fig. 8 Physical implementations of monitoring network routers. (a) Traditional routing; (b) RCA; (c) DBAR

shown in Fig. 8(c).

Ramanujam et al. [35] proposed DAR algorithm which applies flooding to transfer the latency information, and controls the flow by estimating the latency of the candidate nodes. Routers broadcast the stored information of other nodes. Therefore, all neighboring nodes of node C, within an area with radius N, can receive node C's status in N clock cycles at most. The propagation proceeds periodically.

Global adaptive routing algorithms, which are physically implemented by routing table [16,42,43], are similar to the traditional distributed network, and will not be discussed in detail in this paper.

The algorithms above evaluate the congestion status with weight to arbitrate between candidate nodes. Kakoulli et al. [15] proposed a mechanism to predict the network's hotspot with a novel frame of artificial neural networks (ANN).

4.3 Hotspot prediction based on ANN

Hotspot is defined as a group of nodes (usually adjacent ones) or some traffic patterns in the network with relatively frequent communications during a period of time. Hotspots emerge with the workload imbalance. Bypassing these hotspots can ease the imbalance of the network and significantly improve efficiency. In distributed networks, there have been already some promising results on hotspot prediction [37,44]. Different from those selection-based algorithms mentioned previously, hotspot prediction is based on searching.

ANN [45] is a structure formed by plenty of basic modules. These basic modules, which are called neurons, have similar logics. Every neuron takes a real value vector as input, and outputs a real number. Figure 9 is a typical ANN which is divided into input layer, output layer and hidden layer. Theoretically, an ANN with only one hidden layer can approximate any complex function with arbitrarily small bias.

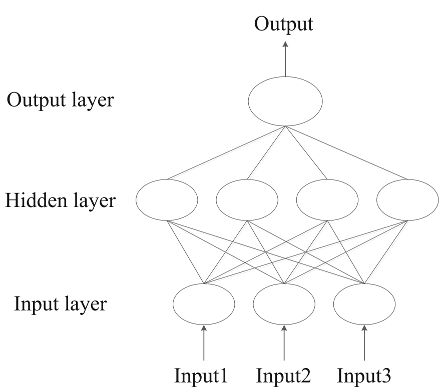


Fig. 9 An ANN with one hidden layer

Kakoulli et al. [15] divided the whole net into several segments in variant scales (2×2/3×3/4×4). Then the congestion status of these segments are regarded as ANN input. Figure 10 is an example of a 4×4 network using 9/4/1 neurons to monitor segments of size $2 \times 2/3 \times 3/4 \times 4$. The congestion metrics derived by the neurons are inputs of ANN, linking to ten nodes from hidden layer, and finally outputting to the output layer.

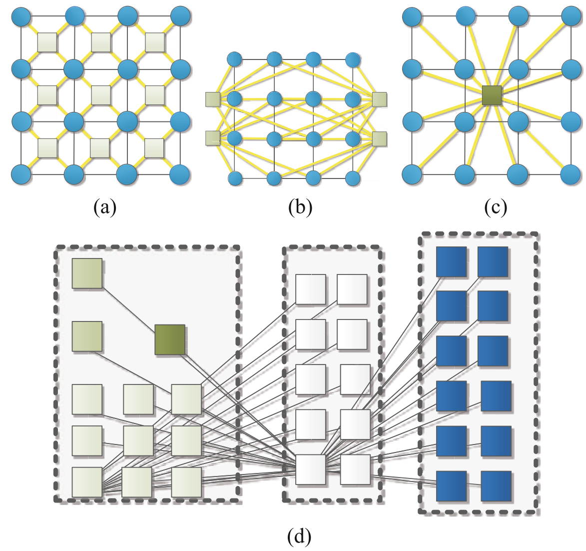


Fig. 10 Physical implementations of monitoring network routers: monitoring network with (a) 2×2 sub-areas; (b) 3×3 sub-areas; (c) 4×4 sub-areas; and (d) corresponding ANN structure (not all the linkages are drawn)

In this example, the inputs are the buffer occupation rates of network nodes, and the output is node which can turn into hotspot with the highest possibility. The researchers use synthetic traffic traces, which contain buffer utilization data collected over 500 000 cycles, as training data. In order to cover variant hotspots' emerging frequency, the training data are derived with the net throughput at 0.22, 0.67 and 0.98. The results indicate that even with a high sampling rate (greater than 60%), the predicting error cannot be limited under 10% (1%–12% usually). The authors state that limiting the predicting error under 5% would be unrealistic. It is also believed that optimizing the sample set could reduce predicting error.

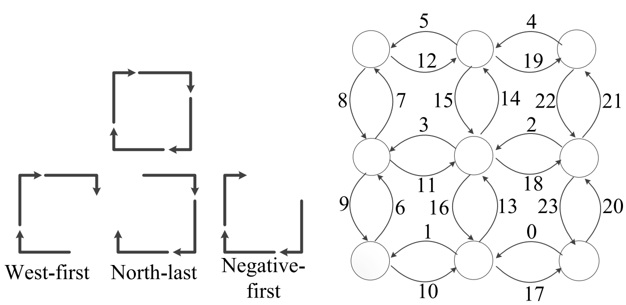
4.4 Deadlock-free theory of adaptive routing algorithm

Adaptive routing algorithms may cause livelocks or deadlocks during the interconnection process. Livelock refers to that the path of data transferring is a closed loop and the destination node cannot be reached. As mentioned previously, livelocks will never emerge in minimal routing algorithms. Either limiting the presence times of non-candidate nodes or ensuring the probability of passing the candidate nodes, can eliminate livelocks in non-minimal routing algorithms [19].

Deadlock refers to a situation that several packets are waiting for each other to release the resources and become a cycle, thus all those packets are blocked. To get rid of deadlock, there are mainly two ways. One is monitoring deadlock in real time and resolving deadlock when it emerges. The other way is restricting the routing algorithms by rules to avoid the emerging of deadlock. The later method is more widely applied because of its simplicity and cheapness. The representatives are Virtual Channel [38] and Turn Model [21].

Duato [22–24] made the fundamental theory for using virtual channel to avoid deadlock: always keeping a vacant channel as escape channel by the routing rules. The core of the theory may be declared as: if there exist a sub-relation R_1 of routing relation R, whose extended channel dependence graph is connected and contains no loop, then R is deadlock-free. In practical implementation, the satisfying R_1 is structured firstly and then the complementary set $R' = R - R_1$ is added.

The turn model [21] regards a ring in mesh as four turnings. There are eight types of turnings (ES, SW, WN, NE in clockwise direction and EN, NW, WS, SE in anti-clockwise direction). Limiting one type of turning for clockwise and anti-clockwise respectively would prevent the occurrence of loops, thus avoiding deadlocks. Three kinds of implementation are proposed based on this principle, namely West-first, North-last and Negative-first, as shown in Fig. 11(a). Figure 11(b) is an example of West-first, passing paths direct west with higher priorities. Turn model is the basis of many other algorithms [46,47] and has been widely applied.



(a) (b)

Fig. 11 Turn model. (a) Three kinds of implementation; (b) an example of West-first

Bolotin et al. [48] proposed another deadlock-free methodology by hardware coding, which, in fact, is equivalent to turn model.

5 Half-adaptive routing algorithm

Designing half-adaptive routing is a tradeoff between algorithm freedom and logic overhead. On the aspect of deadlock-avoidance mechanism, half-adaptive routing limits the number of candidate paths by routing rules such that no virtual channel is needed.

Chiu [25] proposed odd-even turn model based on turn model, to construct a half-adaptive algorithm. The basic concept is as follows. Firstly, columns are labelled and divided into two types, odd and even, according to their order. Then the nodes on the odd columns and even columns are restricted with separate rules, such that turning type EN & NW (also ES & SW) will not emerge in the same column. To be specific, EN and ES turnings are forbidden on even columns, while on odd columns, NW and SW turnings are forbidden. In the paper Chiu proved that such constraints ensure odd-even turn model to be deadlock-free. Compared with turn mode, odd-even turn model restricts less and chooses the path more freely.

Noticing the variant behaviour of oblivious routing and adaptive routing under different network load, Hu and Marculescu [27] proposed DyAD. the main idea of DyAD is using oblivious routing when the workload is light, and using adaptive routing when the workload is heavy. DyAD adds a mode controller to monitor the congestion status of adjacent nodes. The routing algorithm uses XY-routing at normal situations and switches to odd-even turn model when congestion signals are received.

Fu et al. [26] proposed another half-adaptive algorithm named Abacus, which is also based on turn mode. Regarding the $n{\times}n$ mesh as an Abacus of n columns and each columns has n beads. In each column, there is a clockwise bead and a counter-clockwise bead specified. Nodes above clockwise (counter-clockwise) beads forbid ES (NW) turning, and nodes below them forbid SW (EN) turning. This rule ensures no loop would emerge, as shown in Fig. 12.

The adaptiveness of the algorithm lies in that the clockwise/counter-clockwise beads will move according to the network status, depending on the flow/stream of the network on the upper part and lower part of the columns. When the imbalance reaches a threshold, the beads will move accordingly. In the paper, Fu proved that this algorithm is both deadlock-free and livelock-free, and that for each source-destination node pair there are at least two feasible paths.

6 Future work

Routing algorithm is one of the key factors for NoC performance. Great progresses have been made in both adaptive routing and half-adaptive routing. Considering the disadvantages of current algorithms, aspects below are worth further studing.

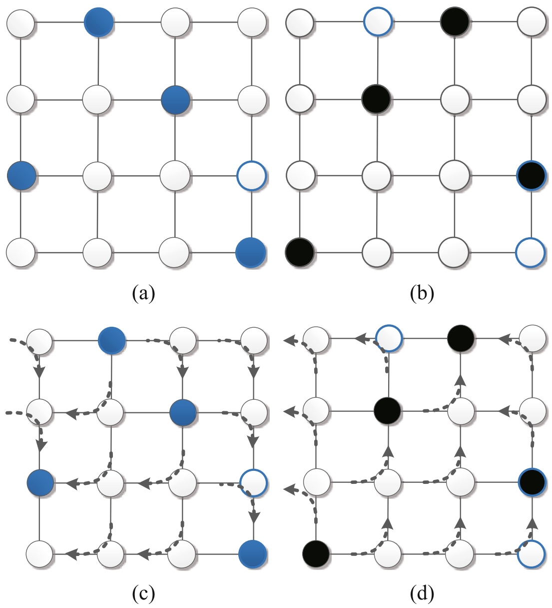


Fig. 12 Abacus. (a) Clockwise beads; (b) counter-clockwise beads; (c) forbidden turns (clockwise); (d) forbidden turns (counter-clockwise)

Firstly, most of the current global adaptive routing algs [10,12–14,19,20] use congestion information of current node or the neighboring nodes, to select which direction should be chosen. The current node is deeply investigated along with the neighboring nodes, while from the aspect of destination node, part of the collected information is irrelevant, e.g., RCA [13]. There is another doubt, when calculating weighted average (e.g., the contention value) in [2,13,14], for convenience of hardware implementation, they all use 1/2 as the weight value. The weight 1/2 might not be optimal, hence theoretical analysis and additional experiments are needed.

Secondly, global adaptive routing algorithms calculate the best path every hop, while in practical the congestion statuses of candidate channels are often quite similar, thus the algorithms cannot avoid hotspot as we expected. Hotspot prediction algorithms show their advantages under these circumstances. Promising research is done on hotspot prediction based on ANN [15], indicating that machine learning methods might have tremendous breakthrough on hotspot prediction, yet the algorithm parameters remains to be optimized and other machine learning methods are expected in the exploration of hotspot prediction.

Thirdly, the physical implementation structures of global adaptive routing algorithms are mostly based on connections between nodes, while the hotspot-related algorithms are controlled by a central module. Central controlled pattern is suitable for global information gathering, while node-to-node connection pattern has fine property on scalability. Designing the appropriate connecting structure according to the features of routing algorithms, is an interesting topic for further study.

Last, half-adaptive routing could be implemented in various ways. Either restricting path selection [21,25] or changing algorithm mode by monitoring network flow [26,27] is a promising method. For the second type, designing an mode controller, which can coordinate with the utilized algorithms accurately, is both important and challenging.

7 Conclusion

In this paper, we introduce the concepts and taxonomy of NoC routing algorithms on mesh topology. Oblivious routing, adaptive routing and half-adaptive routing, each has its own unique features to suit certain real implementation scenarios. The main streams of recent research on mesh NoC are presented and future work are discussed. Considering the trend that more cores will be integrated on chip, NoC will become the major interconnection structure. As a critical factor of NoC performance, routing algorithms will remain to be a research hotspot.

Different from the traditional routing algorithms which lack the necessary information, global adaptive algorithms get more attention because of its robustness under heavy workload. In addition to the traditional point-to-point connection, setting central controlling modules is a promising way to design new global adaptive routing processors. Applying machine learning methods such as ANN may make the network more intelligent and self-adaptive. Half-adaptive routing algorithms, combining the advantage of both oblivious routing and adaptive routing, have their own attraction for the high performance-power efficiency. Routing algorithms on mesh NoC will continue to be a thrilling and challenging field for future research.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 61133004, 61222204, 61221062, 61303158, 61432016, 61472396, 61473275, and 61532016), the 973 Program of China (2015CB358800), the Strategic Priority Research Program of the CAS (XDA06010403, XDB02040009), the International Collaboration Key Program of the CAS (171111KYSB20130002), and the 10 000 talent program.

References

1. Dally W J. and Towles B. Route packets, not wires: on-chip inter-

connection networks. In: Proceedings of the 38th Design Automation Conference. 2001, 684–689

2. Gratz P, Kim C, McDonald R, Keckler S W, Burger D. Implementation and evaluation of on-chip network architectures. In: Proceedings of International Conference on Computer Design. 2006, 477–484
3. Benini L, Micheli D G. Networks on chip: a new paradigm for systems on chip design. In: Proceedings of Design, Automation and Test in Europe Conference and Exhibition. 2002
4. Vangal S, Howard J, Ruhl G, Dighe S, Wilson H, Tschanz J, Finan D, Iyer P, Singh A, Jacob T, Jain S, Venkataraman S, Hoskote Y, Borkar N. An 80-tile 1.28 TFLOPS network-on-chip in 65nmCMOS. In: Proceedings of IEEE International Solid-State Circuits Conference. 2007
5. Sankaralingam K, Nagarajan R, Gratz P, Desikan R, Gulati D, Hanson H, Kim C, Liu H, Ranganathan N, Sethumadhavan S, Sharif S, Shivakumar P, Yoder W, McDonald R, Keckler S, Burger D. Distributed microarchitectural protocols in the TRIPS prototype processor. In: Proceedings of International Symposium on Microarchitecture. 2006, 480–491
6. Liang J, Swaminathan S, Tessier R. aSOC: a scalable, single-chip communication architectures. In: Proceedings of the 22nd International Conference on Parallel Architectures and Compilation Techniques. 2000, 37–46
7. Bjerregaard T, Mahadevan S. A survey of research and practices of network-on-chip. ACM Computing Surveys, 2006, 38(1): 1–51
8. Intel Corporation. A Touchstone DELTA System Description. Technical Report.1991
9. BadrH G, Podar S. An optimal shortest-path routing policy for network computers with regular mesh-connected topologies. IEEE Transactions on Computers, 1989, 38(10): 1362–1371
10. Nesson T, Johnsson S L. ROMM routing on mesh and torus networks. In: Proceedings of Annual ACM Symposium on Parallel Algorithms & Architectures. 1995, 275–287
11. Singh A, Dally W, Gupta A, Towles B. GOAL: a load-balanced adaptive routing algorithm for torus networks. In: Proceedings of the 30th Annual International Symposium on Computer Architectur. 2003, 194–205
12. Li M, Zeng Q A, Jone W B. DyXY—a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In: Proceedings of the 43rd ACM/IEEE Design Automation Conference. 2006, 849–852
13. Gratz P, Grot B, Keckler S. Regional congestion awareness for load balance in networks-on-chip. In: Proceedings of the 14th IEEE International Symposium on High Performance Computer Architecture. 2008, 203–214
14. Ma S, Jerger N, Wang Z Y. DBAR: an efficient routing algorithm to support multiple concurrent applications in networks on-chip. ACM SIGARCH Computer Architecture News, 2011, 39(3): 413–424
15. Kakoulli E, Soteriou V, Theocharides T. Intelligent hotspot prediction for network-on-chip-based multicore systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2011, 31(3): 418–431
16. Dally M J, Towles B P. Principles and Practices of Interconnection Networks. San Francisco:Morgan Kaufmann Publishers Inc., 2003
17. Valiant L G. A scheme for fast parallel communication. SIAM Journal on Computing, 1982, 11(2): 350–361
18. Singh A, Dally W J, Towles B, Gupta A K. Locality preserving randomized routing on torus networks. In: Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures. 2002, 9–13
19. Ascia G, Catania V, Palesi M, Patti D. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. IEEE Transactions on Computers, 2008, 57(6): 809–820
20. Ramanujam R S, Lin B. Destination-based adaptive routing on 2D mesh networks. In: Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems. 2010
21. Glass C, Ni L. The turn model for adaptive routing. In: Proceedings of the International Symposium on Computer Architecture. 1992, 278–287
22. Duato J. A new theory of deadlock-free adaptive routing in wormhole networks. IEEE Transactions on Parallel and Distributed Systems, 1993, 4(12): 1320–1331
23. Duato J. A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks. IEEE Transactions on Parallel and Distributed Systems, 1995, 6(10): 1055–1067
24. Duato J. A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks. IEEE Transactions on Parallel and Distributed Systems, 1996, 7(8): 841–854
25. Chiu G M. The odd-even turn model for adaptive routing. IEEE Transactions on Parallel and Distributed Systems, 2000, 11(7): 729–738
26. Fu B Z, Han Y H, Ma J, Li H W, Li X W. An Abacus turn model for time/space-efficient reconfigurable routing. In: Proceedings of the 38th annual international symposium on Computer architecture. 2011, 259–270
27. Hu J, Marculescu R. DyAD—smart routing for networks-on-chip. In: DAC 2004, 2004, 260–263
28. Marculescu R, Ogras U Y, Peh L S, Jerger N E, Hoskote Y. Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2009, 28(1): 3–21
29. Ogras U Y, Marculescu R. Analysis and optimization of prediction-based flow control in networks-on-chip. ACM Transactions on Design Automation of Electronic Systems, 2008, 13(1): 105–133
30. Barrow-Williams N, Fensch C, Moore S. A communication characterization of splash-2 and Parsec. In: Proceedings of IEEE International Symposium on Workload Characterization. 2009, 86–97
31. Kakoulli E, Soteriou V, Theocharides T. Intelligent hotspot prediction for network-on-chip-based multicore systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(3): 418–431
32. Singh A, Dally W J, Towles B, Gupta A K. Globally adaptive load-balanced routing on tori. Computer Architecture Letters, 2004, 3(1): 6–9
33. Woo S C, Ohara M, Torrie E, Singh J P, Gupta A. The splash-2programs: Characterization and methodological considerations. ACM SIGARCH Computer Architecture News, 1995, 23(2): 24–36
34. Henning J L. SPEC CPU2000: measuring CPU performance in the new millennium. Computer, 2000, 33(7): 28–35
35. Ramanujam R S, Lin B. Destination-based adaptive routing on 2Dmesh networks. In: Proceedings of the 6th ACM/IEEE Symposium on Architecture for Networking and Communications Systems. 2010

36. Sasakawa R, Kise K. LEF: long edge firstrouting for two-dimensional mesh network on chip. In: Proceedings of the 6th International Workshop on Network on Chip Architectures. 2013, 5–10
37. Baydal E, Lopez P, Duato J. A family of mechanisms for congestion control in wormhole networks. IEEE Transactions on Parallel & Distributed Systems, 2005, 16(9): 772–784
38. Dally W J, Aoki H. Deadlock-free adaptive routing in multicomputer networks using virtual channels. IEEE Transactions on Parallel & Distributed Systems, 1993, 4(4): 466–475
39. Ebrahimi M, Daneshtalab M, Liljeberg P, Plosila J, Tenhunen H. CATRA-congestion aware trapezoid-based routing algorithm for on-chip networks. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition. 2012, 320–325
40. Wang J H, Gu H X, Yang Y T, Wang K. An energy-and buffer-aware fully adaptive routing algorithm for network-on-chip. Microelectronics Journal, 2013, 44(2): 137–144
41. Qian Z L, Bogdan P, Wei G P, Tsui C Y,Marculescu R. A traffic-aware adaptive routing algorithm on a highly reconfigurable network-on-chip architecture. In: Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/software Codesign & System Synthesis. 2012, 161–170
42. Wang L, Song H, Jiang Y T, Zhang L H. A routing table-based adaptive and minimal routing scheme on network-on-chip architectures. Computers and Electrical Engineering, 2009, 35(6): 846–855
43. Palesi M, Kumar S, Holsmark R. A method for router table compression for application specific routing in mesh topology NoC architectures. Lecture Notes in Computer Science, 2006, 4017: 373–384
44. Duato J, Johnson I, Flich J, Naven F, Garcia P, Nachiondo T. A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks. In: Proceedings of the 19th IEEE International Symposium on High Performance Computer Architecture. 2005, 108–119
45. Mitchell T. Machine Learning. New York: McGraw-Hill Inc., 1997
46. Kumar M, Laxmi V, Gaur M S, Ko S B, Zwolinsk M. CARM: congestion adaptive routing method for on chip networks. In: Proceedings of International Conference on VLSI Design. 2014, 240–245
47. Kumar M, Laxmi V, Gaur M S, Daneshtage M,Zwolinsk M. A novel non-minimal turn model for highly adaptive routing in 2D NoCs. In: Proceedings of the 22nd International Conference on Very Large Scale Integration. 2014, 1–6
48. Bolotin E, Morgenshtein A, Cidon I, Kolodny A. Automatic and hardware-efficient SoC integration by QoS network on chip. In: Proceedings of the 11th IEEE International Conference on Electronics, Circuits & Systems. 2005, 479–482

Yue Wu received the BS degree in statistics from University of Science and Technology of China, China in 2006. He is currently a PhD candidate of Institute of Computing Technology, Chinese Academy of Sciences, China. His main research interests include computer architecture, computational intelligence and network on chip.

Chao Lu received the BE degree in EE from University of Science and Technology of China, China in 2006. He is currently a PhD candidate of Institute of Computing Technology, Chinese Academy of Sciences, China. His main research interests include computer architecture, and network on chip.

Yunji Chen was graduated from the Special Class for the Gifted Young, University of Science and Technology of China, China in 2002. He received the PhD degree in computer science from Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), China in 2007. He is currently a professor at ICT, CAS. He was the awardee of the NSFC Excellent Young Scholars Program in 2012. His research interests include parallel computing, microarchitecture, hardware verification, and computational intelligence.