FT-LTR: Low-Timedelay Fault-Tolerant Algorithm for Mesh based Network-on-Chip

Xiucheng Wang and Longfei Ma

Abstract—**We explore fault-tolerant routing methods for mesh based Network-on-Chip by using dynamic switch XY-YX routing and generate fault nodes area. We begin by analyzing the difficulty of traditional dynamic XY-YX algorithms, and found that as the number of fault nodes is increasing, they are more likely to make wrong dicision. Thus, we improves a new algorithm to solve this problem. Then we analyze the delay caused by our algorithm, and propoed the conception of useless nodes to decrease time delay.**

Index Terms—**fault-tolerant routing, Network-on-Chip, dynamic, delay, useless nodes.**

I. Introduction

With the development of chip fabrication, the size of the chip is shrinking while the number of its cores are increasing, hence problems as natural aging are becoming gradually prominent[1][2].Therefore, in order to prolong the life-span of the chip, the NoC-based fault-tolerant algorithm rises in response[3].The NoC-based fault-tolerant routing algorithm is designed to improve the data distribution capability of the NoC system in the presence of faulty nodes or faulty links in the network, thus improving system reliability and robustness. Since the current NoC architecture mostly uses 2D mesh topology, the 2D-mesh-based fault-tolerant routing algorithm has always been a key point in routing algorithm design. Compared with the traditional method of Dimension-Ordered Routing, the adaptive routing algorithm shows a huge advantage in solving network congestion, reducing delay, and especially improving the fault tolerance of routing[2]. In this research work, the dynamic XY algorithm which is based on fault section detection is the emphasis of this paper. The fault section detection can reduce the algorithm complexity in the routing process, while the dynamic character of this algorithm can improve utilization efficiency of the nodes and have more salient time-delaying character.

In summary, our main contributions are:

- Improve the reliability of the fault-tolerant algorithm and increase the success rate of routing.
- Efficiently deal with the fault tolerance problem in the state where there are many failed nodes and the distribution is concentrated.
- Reduce transmission delay during fault-tolerant routing.
- Increase node utilization in the case of generating the necessary fault areas.

Xiucheng Wang and Longfei Ma are with the School of Telecommunications Engineering, Xidian University.

II. Related Work

Since the fault is inevitable due to natural oxidation, high-energy ray, temperature and humidity changes during the life cycle of the chip, how to ensure that the basic functions of the chip are not affected in the event of a local failure has always been the direction that researchers are focusing on[3].

In recent years, the work of fault-tolerant routing algorithms for mesh can be divided into the following aspects: 1) dynamic switching of XY and YX algorithms to achieve the avoidance of failed nodes; 2) using nodes which directly connect with the failed nodes to realize the bypass routing ; 3) Generating fault sections to achieve fault tolerance. In the work of Khichar et al., the dynamic XY-YX algorithm is applied. When there is no fault in the network, the XY algorithm is used for routing. If a fault occurs during routing along the X dimension, It is switched to the Y dimension and transferred to the destination node. Similarly, if a failure is encountered during transmission in the Y direction, the data is transmitted in the X direction until the route is successful[4].The work of Sinha et al.further improved the XY-YX dynamic switching algorithm and made a brief and clear definition of how to deal with the situation where all three links around the node are infeasible [5].

In the study of the use of bypass routing, Priya et al. realized the use of bypass nodes to achieve fault avoidance[6] by specifying the direct and indirect neighbor nodes. In the algorithm based on the fault region generation, Krishnan's work innovatively define the unreachable and unavailable nodes and marked most of them with non-rectangular regions, instead of the traditional generation method of fault loops and fault chains[7]. In addition, they utilizes the generation of the unreachable and unavailable nodes is utilized so that the transmission delay is reduced while the fault tolerance is also realized[7].

III. Proposed Algorithm

This section will mainly discuss the implementation principle of the new fault-tolerant routing algorithm proposed by us, and its advantages compared with the alternative algorithms.

A. Low Time Delay Fault-Tolerant Algorithm

Our algorithm is based on the XY routing algorithm. When the data starts to be routed, the XY algorithm is used for routing until the route is successful or encounters a failed node.

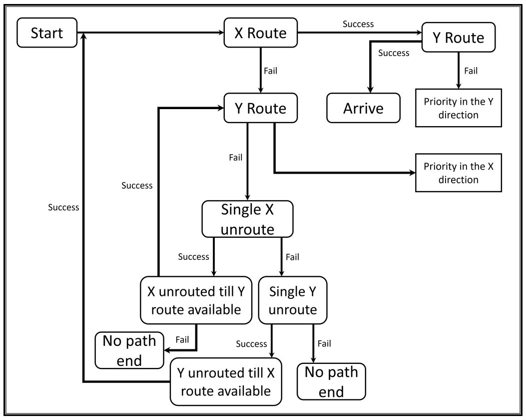


Fig. 1: Flowchart of the proposed algorithm

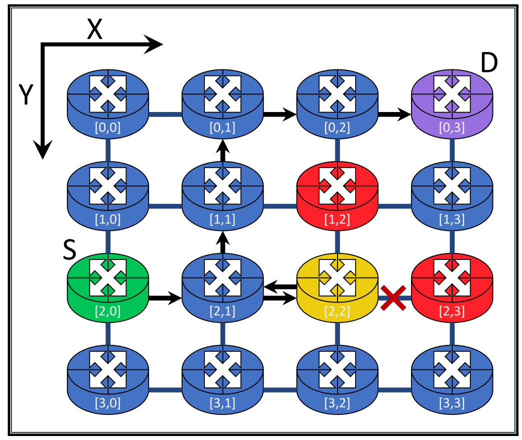


Fig. 2: Unroute will increase the number of hop

When encountering a failed node, the algorithm will implement a series of methods such as switching the routing direction to realize fault-tolerant routing with minimal skipping for the data packet, at the same time avoiding deadlocks. That is, in the process of data transmission, if a failed node is encountered, the routing direction is switched to avoid the fault in the NoC. If more faults are encountered, it needs to be discussed separately.

Fig 1 details the algorithm principle in the form of a flowchart. If a packet encounters a failed node during routing along the x-axis, the data is propagated in the direction of the y-axis near the target node. If the packet encounters a failure during routing along the y-axis, the data is propagated in the x-axis direction near the target node. If the data encounters a fault during routing along the x-axis and immediately encounters a fault after switching to the y-direction, the data is routed one step away from the target node along the y-axis and then routed along the x-direction. If it is found that the x-direction route is not feasible at this time, it continues to follow the direction of the y-axis away from the target node until the link, which locates at the x-direction of the node of the packet and is close to the target node, is feasible. Then the route along the y-axis is ended and shifted to the x-axis.

If there is no link available in the x-axis direction during propagation along the y-axis away from the target node, and the failed link is also encountered on the y-axis, the route fails. Similarly, in the process of y-axis routing, when encountering faults in the x and y directions at the same time, the data is propagated along the x-axis away from the target node until an available link appears on the y-axis. In particular, if three failed nodes are encountered in the process of x routing and can only be routed along negative direction of the x-axis, then one step is reversed along the x axis, and then the data is routed along the y axis to the destination node. .If the node where the data packet is located is in the same row or in the same column as the target node, routing is performed according to the priority in each direction.The pseudocode is show in Algorithm 1.

B. Further Improvement

In further work, we found that as the number of faults in the NoC increases, it not only reduces the success rate of the route, but also increases the time delay. The increase of delay mainly comes from the fact that the algorithm needs to perform reverse routing in the process of fault-tolerant decision-making, that is, routing to the direction away from the target node to achieve fault avoidance. As shown in Fig 2, the traditional fault-tolerant algorithm often reverses routing when [1, 2] and [2, 3] are not feasible, so that the data can reach the destination node. From the figure we can also clearly see that the entry of data from [2,1] to [2,2] not only has no positive effect on the final routing effect, but will return [2,1] from [2,2] and add two unnecessary steps.

In that sense, we can define a node like [2, 2] as an invalid node, that is, entering this node can not play a positive role in the final routing result. For the generation of useless nodes, we refer to the traditional fault section generating algorithm and the definition of unreachable nodes and unavailable nodes[7] put forward by Krishnan et al. By simplificating, we define the useless nodes as follows: if the northbound node and the eastbound node are both failed nodes or northeast invalid nodes, then the node itself is marked as northeast invalid node. It means that if the destination node is in the northeast direction of the current node, the data packet will not enter the node preferentially during the transmission process, but will be processed in a manner similar to the failed node.

In the case of generating an invalid node, the routing path is as shown in Fig 3. First, the data packet starts from [2, 0], and is transmitted to the [2, 1] according to the XY algorithm under the initial state. In the process of continuing the route, the invalid node [2, 2] is encountered and converted to the YX algorithm, which is routed north along the y axis to [0, 1], and then routed to the destination node along the x direction, and the route ends. It can be seen that the geometric distance between S and D is 5, and the total number of hops of the route is also 5, so the introduction

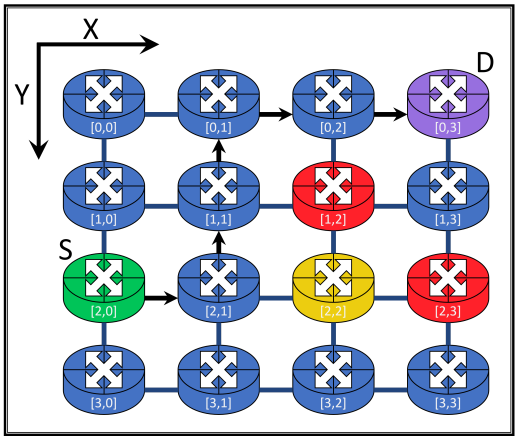


Fig. 3: Useless nodes help decrease the number of hop

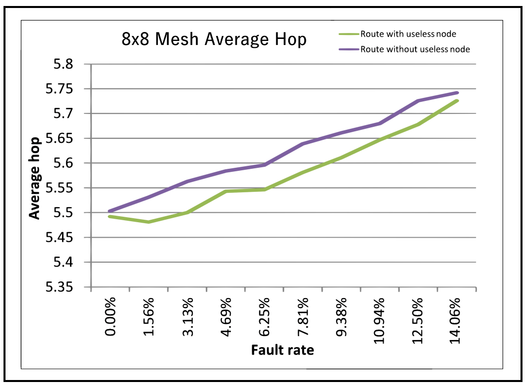


Fig. 4: Average hops with increasing fault rate

of invalid nodes reduces hops during the transmission on the basis of ensuring the fault tolerance performance of the algorithm, thereby decreasing time delay of the transmission.

IV. Experiments Result

Now we will show the routing reliability of the algorithm both under the situation where the fault of the network increases with a certain mesh size, and the circumstance where the size of mesh increases with certain fault percentage.

A. Routing Success Rate

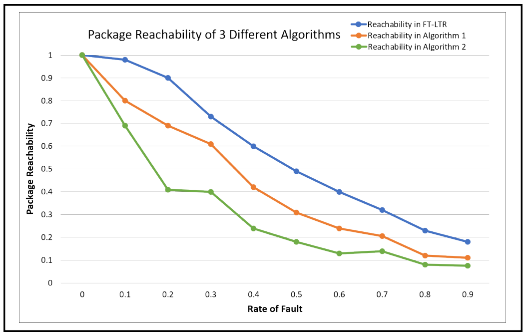
First, in the 4*4 mesh structure, we randomly set the target node and the fault node, gradually increase the number of faults from zero, and compare it with the previously proposed algorithm. It can be seen from Fig 5(a) that although the accessibility of the algorithm is reduced as the number of faults increases, it is higher than the previously proposed algorithm. Especially when the number of faults reaches about 20%, the algorithm still has a success rate of about 90%.Then we test the success rate of the algorithm in different sizes of mesh structures with a fixed failure rate of 10%. It can be seen that as the network size increases, the route reachability remains basically the same, and in the process of changing from a smaller network to a larger network, the route reachability will slightly increase.In addition, we also analyze the impact of introducing invalid nodes on the routing reliability. The results show that when the failure rate is higher than a certain value, introducing the concept of invalid nodes will help improve routing reliability.

Algorithm FT-LTR 1

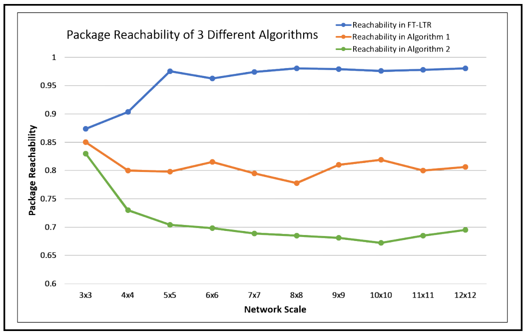
define X_s, Y_s — destination x and y coordinates
define X_c, Y_c — current x and y coordinates.
define X_d, Y_d — $|X_s - X_c|$, $|Y_s - Y_c|$
define $Xrouting$ — route to decrease X_d
define $Yrouting$ — route to decrease Y_d
define $Xunrouting$ — route to increase X_d
define $Yunrouting$ — route to increase Y_d
function Routing(X_d, Y_d)
  if $X_d == 0\&\ Y_d == 0$ **then**
    return $Success$
  else if $X_d! = 0\&\ Y_d! = 0$ **then**
    if (**then**X routing available)
      route X
    else if Y routing available **then**
      route Y
    else if X unrouting avaliable **then**
      repeat
        X unrouting
      until Y routing available
      $then$
      Yrouting
      if Y routing always unavaliable **then**
        return FAIL
      end if
    else if Y unrouting available **then**
      repeat
        Y unrouting
      until Y routing available
      $then$
      Xrouting
      if X routing always unavaliable **then**
        return FAIL
      end if
    else
      return FAIL
    end if
  else if $X_d == 0\ ||Y_d == 0$ **then**
    priority in the x or x direction
  end if
end function

B. Time Delay

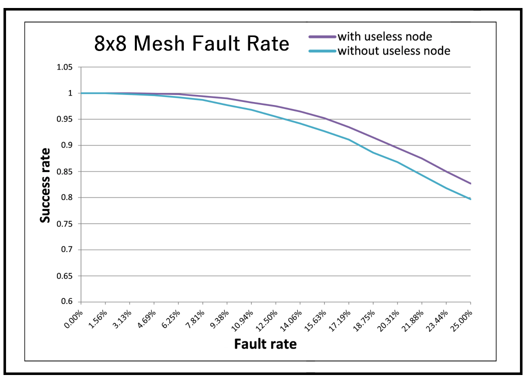
Since the concept of invalid node is introduced, as long as there is a propagation path the routing steps of which equals to the direct geometric distance between the target node and the source node, the algorithm can be used to find the path with great probability, thereby reducing the



(a) With increasing fault-node rate



(b) With increasing size of mesh



(c) With useless nodes

Fig. 5: Success rate in different situation

propagation delay. Fig 4 show the average hops for routing of our algorithm is always low than other algorithms. Fig 5(b) shows that as the number of faults in the network increases, though the number of hops required to reach a successful route will gradually increase, it can be clearly seen that the introduction of invalid nodes will help reduce the number of hops needed for routing, comparing the two data.

V. Conclusion

This paper proposes a new type of fault-tolerant routing algorithm. Simulation proves that the algorithm's fault tolerance rate and average hop are lower than the existing algorithms.

References

[1] L. Benini and G. De Micheli, "Networks on chips: A new soc paradigm," *computer*, vol. 35, no. 1, pp. 70–78, 2002.

[2] J.-S. Wang and L.-T. Huang, "Review on fault-tolerant designs for network-on-chip," *Journal of Electronic Science and Technology*, pp. 1–20, 2018.

[3] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, p. 1, 2006.

[4] J. Khichar, S. Choudhary, and R. Mahar, "Fault tolerant dynamic xy-yx routing algorithm for network on-chip architecture," in *2017 International Conference on Intelligent Computing and Control (I2C2)*. IEEE, 2017, pp. 1–6.

[5] D. Sinha, A. Roy, K. V. Kumar, P. Kulkarni, and J. Soumya, "D n-ftr: Fault-tolerant routing algorithm for mesh based network-on-chip," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*. IEEE, 2018, pp. 1–5.

[6] S. Priya, S. Agarwal, and H. K. Kapoor, "Fault tolerance in network on chip using bypass path establishing packets," in *2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID)*. IEEE, 2018, pp. 457–458.

[7] H. Krishnan and L. Preethi, "Fault-tolerant routing with minimum delay for mesh architecture," in *2018 International CET Conference on Control, Communication, and Computing (IC4)*. IEEE, 2018, pp. 322–325.