ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

EFFICIENT FAULT AWARE ROUTING IN NETWORK ON CHIP SYSTEMS

Ms. M. KEIRTHI, IInd ME (Communication System),
Dr. S. KARTHIGAI LAKSHMI, Ph.D., Professor/ECE
SSM Institute of Engineering and Technology, Dindigul, Tamilnadu, India
keirthi.manoharan@gmail.com

ABSTRACT

The advanced deep sub micrometer technology increases the risk of failure for on-chip components. In advanced Network-On-Chip (NoC) systems, the failure constrains the on- chip bandwidth and network throughput. Fault-tolerant routing algorithms aim to alleviate the impact on performance. Some fault aware techniques have integrated the congestion, deadlock, and fault-awareness information in channel evaluation function to avoid the hotspot around the faulty router. To solve this problem, the ant colony optimization-based fault-aware routing (ACO-FAR) algorithm for load balancing in faulty networks is proposed. The behavior of an ant colony while facing an obstacle (failure in NoC) can be described in three steps: 1) encounter; 2) search; and 3) select and the corresponding mechanisms are: 1) notification of fault information; 2) path searching mechanism; and 3) path selecting mechanism. With proposed ACO-FAR, the router can evaluate the available paths and detour packets through a less-congested fault-free path. The proposed algorithm provides higher throughput. In addition, it balances the distribution of traffic flow in the faulty network.

1. Introduction

Networking is the practice of linking multiple computing devices together in order to share resources. These resources can be printers, CDs, files, or even electronic communications such as e-mails and instant messages. These networks can be created using several different methods, such as cables, telephone lines, satellites, radio waves, and infrared beams. Without the ability to network, businesses, government agencies, and schools would be unable to operate as efficiently as they do today [10]. The ability for an office or school to connect dozens of computers to a single printer is a seemingly simple, yet extremely useful capability. Perhaps even more valuable is the ability to access the same data files from various computers throughout a building. This is incredibly useful for companies that may have files that require access by multiple employees daily. By utilizing networking, those same files could be made available to several employees on separate computers simultaneously, improving efficiency.

Network on Chip (NoC or NOC) is a communication subsystem on an integrated circuit (commonly called a "chip"), typically between intellectual property (IP) cores in a system on a chip (SoC). NoCs can span synchronous and asynchronous clock domains or use unclocked asynchronous logic.

2. Related Work

With the development of Moore’s law, more and more cores can be integrated into a single chip. Future chips may even contain hundreds of cores, in which cases traditional bus interconnections may have to be replaced by NoC in order to support the large number of cores [5]. When designing future many-core processors, devising energy-efficient, high-throughput on chip networks will become one of the great challenges [6], [4], [8].

All Rights Reserved © 2016 IJARBEST

Routing algorithm plays an important role in No C. Early studies suggested a number of oblivious routing algorithms which completely determines the path of packet given the source and destination. Dimension-order routing (DOR) specifies in advance an order between dimensions of a network, and it routes packets strictly with the order. Zigzag routing chooses the dimension having the farthest distance to the destination as the output dimension. ROMM randomly inserts some intermediate nodes between source and destination and the routing between intermediate nodes follows a fixed dimension order. While simple and straightforward, oblivious routing are sometimes ineffective on balancing the network load.

To address the above concern, adaptive routing algorithms have received extensive investigations over the past decade. They are acknowledged to be promising and practical [7] as they leverage dynamic network statues to control the routing. Depending on the congestion awareness in the spatial scale, adaptive routing can be categorized into two types, local adaptive routing and non local adaptive routing. Local adaptive routing only utilizes the congestion information of local links to make the routing decision. LOCAL makes a routing decision by comparing congestion information of local links. NoP further considers congestion information of nodes adjacent to neighbors when making a routing decision. As local adaptive routing techniques only consider local information, they are deemed to be short-sighted [11] and might make inappropriate routing decision.

To alleviate the above burden, recent studies switched to non-local adaptive routing, which employs additional hardware dedicated to propagating non-local congestion information among nodes. RCA introduces an additional CPN (16/32 bits per pair of adjacent nodes) to make each node aware of congestion information of distant links [11]. RCA has to mix up congestion information of different distant links to save wiring cost, thereby often unnecessarily takes unrelated links into account, which may lead to inappropriate routing decisions.

Destination-Based Adaptive Routing (DBAR) employs a different CPN, which effectively filters information of unrelated links [1]. CPN of DBAR only connects nodes along the same dimension to reduce the wiring cost, thereby each node is not aware of congestion information of links that are in neither the same row nor the same column of the node. An example, where node C(5,5) does not know the congestion knowledge of links not in row 5 and column 5 when adopting DBAR. In fact, congestion information of such links was considered to be helpful to the output link selection [11], which has again been validated by our experimental study on the proposed Free- Rider. Moreover, Ramanujam and Lin proposed a CPN which can broadcast congestion information of each node to all nodes [3]. Each round of broadcasting takes O(N logN) cycles, which produces a negative impact on the scalability of the technique. Jiang et al. proposed an adaptive routing technique on dragonfly topology, which broadcasts congestion information of nonlocal links to all adjacent routers by sending dedicated flits for propagating congestion information every cycle when a channel is idle, and also piggybacks congestion information on 1 percent of all packets. Significantly different from the above techniques, Free-

Rider is a non-local adaptive routing technique that does not introduce additional wire/flit for propagating nonlocal congestion information. The most notable innovation of FreeRider is that it leverages free bits in head flits of existing packets to carry and propagate congestion information. From the algorithmic perspective, head flits have sufficient free bits to carry richer information than CPN does, leading to two advantages of FreeRider in comparison with traditional non-local adaptive routing techniques.

First, FreeRider is not disturbed by unrelated links, as statuses of distant links no longer need to be mixed up to save wiring cost. Second, precise congestion information of more distant links can be carried and propagated to benefit the output link selection. Recently, Samman et al. proposed a novel adaptive routing selection strategy [9] which leverages the number of free reservable ID slots and bandwidth space occupancy to make runtime routing decisions for NoC.

Compared with traditional queue-length-oriented adaptive routing selection strategies, the proposed strategy does not suffer from unpredictable traffic situations. In the experiments of FreeRider, we also observe that, the congestion information may change when/after a routing decision is made. The corresponding impact on the network performance of FreeRider is trivial, because FreeRider leverages hotspot status rather than fine-grained congestion status to make routing decisions.

Different from fine-grained congestion status which may rapidly change, hotspot status often keeps unchanged for a long period in practice. Meanwhile, in the routing selection strategy proposed by Samman et al., it only leverages local free reservable ID slots and bandwidth space occupancy information. FreeRider takes into account both local congestion information and non-local hotspot status to make routing decisions.

Ramakrishna et al. presented a non-local adaptive routing algorithm which is named as GCA [2]. There are two main differences between FreeRider and GCA. First, the two methods carry different congestion information on a packet at every hop. In GCA, at every hop, each node appends congestion information and input link information into the traffic vector field of the head flit. As a result, one node can be aware of the congestion information of a distant link only when it receives a packet which has passed through the distant link. In FreeRider, each node reloads the free bits field of the head flit to carry all congestion information of the k backyard with respect to the output link. Therefore, each time a node receives a packet, congestion information of all links in k-backyard with respect to the input link will be updated. In this sense, FreeRider propagates congestion information of distant links more efficiently than GCA.

Second, GCA and FreeRider employ significantly different strategies on output link selection. GCA picks the path with least congestion from the current node to destination node with a multi-step Dijkstra algorithm. For a mesh with N N nodes, the maximum number of steps in the algorithm is 2N 3. Since different steps of the Dijkstra algorithm can only be sequentially executed, and each step involves multiple addition and comparison operations, it is hard to extend the algorithm to largescale networks. To address the scalability issue of GCA, Ramakrishna el al. proposed Limited GCA for large networks which only maintains congestion information for a limited window of j hops around the current node. Although LGCA partially alleviates the scalability burden of GCA, it simultaneously reduces the network performance. In contrast, FreeRider employs a novel strategy select the output link, which only needs to compare two selection metrics of relevant

3. Ant Colony Optimization Aco Based Fault Aware Routing

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

The ACO-based adaptive routing algorithm, with reinforcement learning behavior to detour from the congestion/fault region, is first developed by AntNet. The difference with other evolutionary algorithm is that the ACO-based routing directly maps the distributed mobile agent-based ants as transmitting packets and the routes as the routing paths to solve control and routing problems in telecommunication and networking. In addition, ACO uses distributed agents to make on-line adaptive routing decisions.

In ACO-based routing, the foraging behavior is used to model the ants in discovering the shortest/less-congested path to a food source and to share that information with the other ants through stigmergy [10]. This is achieved by laying a chemical substance called pheromone that induces changes in the environment which can be sensed by other ants.

In applying ACO in network routing and load-balancing, an artificial ant can be realized as a simple program consisting of simple procedures that simulate the laying and sensing of pheromone. An artificial ant migrates from node to node and emulates the laying of pheromone by updating the corresponding entry in the routing table in a node.

To solve the faulty-network issues for 2-D-mesh NoC, multiple optimization factors are considered in this paper, which includes congestion-awareness, deadlock-awareness, and fault awareness. To the best of our knowledge, few works have discussed on integrating multiple factors in the routing table. The adjustable chemical substance, pheromone, in the routing table of ant colony optimization (ACO) is suitable for integrating these factors. This technique in the NoC system show that the inherent detour behavior of ant colony, and integrate the congestion and deadlock-awareness information to achieve a deadlock-recovery-based fully-adaptive routing.

ACO based FAR algorithm is inspired by the fault-tolerant behavior of ant colony, which consists of three steps: 1) encounter; 2) search; and 3) select. Initially, the ants follow the original pheromone trail, and cannot proceed forward because of the obstacle. Then, they search for other available paths in arbitrary directions to detour from the obstacle. After a short period of discovering, more and fresher pheromone accumulates on the shorter detour. As a result, the ants would select this new path to bypass the obstacle.

1) **Notification Mechanism of Fault Information**: We propose a low-cost dynamic notification mechanism to collect and propagate the fault information to neighboring region of the faulty node in the network.

2) **ACO-Based Fault-Aware Path Searching Mechanism**: To identify legal routing path based on the fault information and provide as much path diversity as possible; this mechanism searches for possible paths to neighboring nodes except for faulty paths.

3) **ACO-Based Fault-Aware Path Selecting Mechanism**: To evaluate the congestion condition of the routing paths and relieve the traffic congestion around the faulty nodes, this mechanism integrates the congestion and regional fault information into ant pheromone to select the better path.

. With the proposed techniques, ACO-FAR improves the fault-tolerance ability of network by reducing the undelivered packet ratio from 14.9% to 0.02% compared with Modified X-First. The system throughput outperformed related works by 29.1%–66.5%. In addition, we also evaluate the area efficiency and energy consumption of the fault tolerant schemes. ACO-FAR has the highest area efficiency and moderate energy consumption.

4. ACO Based Fault-Aware Paths

4.1. Searching Mechanism

To identify legal routing path based on the fault information, the router itself can check the RFI signals and determine whether its adjacent router is normal or faulty. The path searching mechanism provides the candidate output channels based on this path information.

The path searching mechanism searches for all paths to adjacent nodes except for faulty paths to provide higher path diversity. Base on the path information when the packet is being sent from the current router to the next hop, there are three cases as follows. Case I and Case II belong to 1) at least one minimal path is feasible and Case III belongs to 2) when only nonminimal paths are feasible.

Case I. Fully Minimal Path: The path searching mechanism provides fully-minimal paths as candidate channels. In this case, the faulty router is not adjacent to the current router.

Case II. Prohibited Faulty Path: In this case, the faulty router is adjacent to the current router. The candidate channel provided by the path selecting mechanism is North and meanwhile the pheromone of East channel is set to zero.

Case III. Nonminimal Path: The only minimal path from current router to destination router is blocked by the faulty node. Hence, the path searching mechanism provides nonminimal paths for candidate channels instead of interrupting the packet transmission. The situation is called packet detour and is similar to the search behavior of ants.

A Special Case of Case I: In this case, the source/current router receives the same RFI value from the North and East channel, the packet has a chance to be routed toward these two directions. If the routing algorithm chooses the North Channel as the output channel, the packet will be blocked by the faulty node afterwards because of the property of minimal routing. The following detour process causes extra packet transmission latency and congestion. To avoid this blockage, we define the x- and y-axis distances from current router $R_{current}$ to destination router R_{dest} as Δdx and Δdy, respectively.

$$\Delta dx = Distance_{x} (R_{current}, R_{dest})$$

$$\Delta dy = Distance_{y} (R_{current}, R_{dest}). \qquad (1)$$

It is observed that the above mentioned blockage condition happens when: 1) the channels are receiving the same RFI value and 2) either Δdx or Δdy is equal to 0. Hence, an additional constraint is added in the routing function that avoids Δdx or Δdy to reach 0 unless they are both equal to 1. The additional constraint is

$$ch = dir_{x}, \text{ if } (RFI_{x}= RFI_{y} \& ?d_{x} \geq 2 \& ?dy = 1)$$
$$ch = dir_{y}, \text{ if } (RFI_{x}= RFI_{y} \& ?d_{y} \geq 2 \& ?dx = 1)$$

$$ch = (dir_{x},dir_{y}), \text{ if } (RFI_{x}= RFI_{y} \ \& \text{ else}). \qquad (2)$$

ch denotes the candidate output channel. ch is constrained to x/y direction if either Δdy/Δdx is equal to 1.

4.1. Selecting Mechanism

The faulty router essentially sets a limitation on the surrounding bandwidth. The congestion and fault information are integrated in the process of path selection through the use of ACO pheromone table in a new way.

The path selecting mechanism chooses the better output channel by taking the RFI value into consideration for reducing the probability of selecting path toward the faulty region.

Since the fault may cause its nearby region to be congested, the output channel with higher RFI value represents a restriction on the path diversity. Therefore, the fault penalty factor, β, is introduced to the state transition rule when making the selection decision, but this factor does not alter the pheromone table, and it is defined as

$$\beta j = 2^{-RFIout,j}. \qquad (3)$$

β is decided by exponential decay of RFI from channel j. The fault pheromone $Ph_{f}(j,R)$ can be computed by the evaluation metric as follows:

All Rights Reserved © 2016 IJARBEST

$$Ph_f\,(j,R) =(Ph(j,R) + \alpha L_j/1 + \alpha(N_k - 1))\times \beta_j. \qquad (4)$$

The right side of the evaluation metric is multiplied by β of the output channel j. Because the Ph(j,R) in table is also altered by the pheromone evaporation mechanism avoid the formation of deadlock. Thus, the fault pheromone $Ph_f(j,R)$ is now the synthesis of congestion, deadlock-aware and fault-aware information. The selection decision is made by evaluating $Ph_f(j,R)$. This mechanism can reduce the probability of the packet transmission to the congested/faulty region.

5. Performance Analysis

The performance of fault-tolerant routing algorithm degrades with the increase in the number of faulty router. To overcome this, ACO-FAR algorithm was proposed. This algorithm provides appropriate settings for the number of consecutive detouring and the proposed mechanism can achieve successful routing for each source/destination pair in a connected routing region. When this ACO-FAR method is used, the source and destination are selected by default in the particular region and with the available details with regard to the available alternative paths and the faults assigned it selects the right one. In this length is set to 5 and number of fault is set to 2. Packet Delivery Ratio ACO-FAR has much better ability to deliver the packets in the faulty network, because of its fully adaptive routing function that provides higher path diversity to tolerate the fault and it can also improve the packet delivery ratio. The reduced latency can be evaluated using this formula,

$$Latency = Trouter \times H + Tcongestion. \qquad (5)$$

The packet latency is equal to router processing time T_{router} multiplied by hop count H and added by the congestion delay $T_{congestion}$. Because of this reduced latency area efficiency is improved, energy consumption is reduced and consequently throughput is increased.

6. Conclusion and Future Work

ACO based Fault Aware Routing algorithm has been proposed to tolerate on-chip failures, increase network throughput, and decrease total latency of network. The proposed algorithm provides higher path diversity and fault-awareness. By using ACO-FAR, highest performance on both throughput and area efficiency was achieved and also energy consumption was reduced. ACO-FAR also balance the distribution of traffic flow in the faulty network. In future, the latency can be reduced and the area efficiency and throughput be further improved by using GSA algorithm.

REFERENCES

[1] S. Ma, N. Enright Jerger, and Z. Wang, “DBAR: An efficient routing algorithm to support multiple concurrent applications in networks- on-chip,” in Proc. 38th Annu. Int. Symp. Comput. Archit., 2011, pp. 413–424.

[2] M. Ramakrishna, P. Gratz, and A. Sprintson, “GCA: Global congestion awareness for load balance in networks-on-chip,” in Proc. IEEE/ACM 7th Int. Symp. Netw. Chip, 2013.

[3] R. Ramanujam and B. Lin, “Destination-based adaptive routing on 2d mesh networks,” in Proc. IEEE/ACM Int. Symp. Archit. Netw. Commun. Syst., 2010, pp. 1–12.

[4] G. Kim, J. Kim, and S. Yoo, “Flexibuffer: Reducing leakage power in on-chip network routers,” in Proc. ACM/EDAC/IEEE 48th Des. Autom. Conf., 2011, pp. 936–941.

[5] D. Sanchez, G. Michelogiannakis, and C. Kozyrakis, “An analysis of on-chip interconnection networks for large-scale chip multiprocessors,” ACM Trans. Archit. Code Optimization, vol. 7, no. 1, article 4, pp. 1–28, 2010.

[6] A. Bakhoda, J. Kim, and T. Aamodt, “Throughput-effective onchip networks for manycore accelerators,” in Proc. IEEE/ACM 43rd Annu. Int. Symp. Microarchit., 2010.

IJARBEST
Research at its Best

All Rights Reserved © 2016 IJARBEST

[7] D. Chen, N. A. Eisley, Y. Sugawara, S. Kumar, V. Salapura, D. L. Satterfield, B. Steinmacher-Burow, and J. J. Parker, “The IBM blue gene/q interconnection network and message unit,” in Proc. Int. Conf. High-Perform. Comput., Netw., Storage Anal., 2011.
[8] H. Kim, G. Kim, and J. Kim, “Scalable on-chip network in power constrained manycore processors,” in Proc. Int. Green Comput. Conf., 2012.
[9] F. Samman, T. Hollstein, and M. Glesner, “Runtime contention and bandwidth-aware adaptive routing selection strategies for networks-on-chip,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 7, pp. 1411–1421, Jul. 2013.
[10] sShaoli Liu, Tianshi Chen, Ling Li, Xi Li, Mingzhe Zhang, Chao Wang, Haibo Meng, Xuehai Zhou and Yunji Chen, “FreeRider: Non-Local Adaptive Network-on-Chip Routing with Packet-Carried Propagation of Congestion Information”, IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 8, August 2015
[11] P. Gratz, B. Grot, and S. Keckler, “Regional congestion awareness for load balance in networks-on-chip,” in Proc. Int. Symp. High Perform. Comput. Archit., 2008, pp. 203–214.

IJARBEST
Research at its Best !!!