International Journal of Grid and Distributed Computing
Vol. 11, No. 9 (2018), pp.33-50
http://dx.doi.org/10.14257/ijgdc.2018.11.9.04

Design and Implementation of a New Routing Algorithm for Fault Tolerance in Networks on Chip - DINRA-FTNoC

Chakib Nehnouh[1,*] and Mohamed Senouci[1]

[1]University Oran1 Ahmed Ben Bella, Oran, Algeria
{nahnouhc, msenouci}@yahoo.fr

Abstract

The possibility to integrate more and more cores on the same chip puts severe constraints on the reliability, to which it is important to provide correct services in the presence of faults. The system should work properly when some faults occur in routers/links. However, the communication has a huge impact on the performance of the Network on Chip, and designing an efficient routing algorithm is more required. To handle the communication requirements, the routing algorithm should find a new path to steer packets from the source to the destination in a faulty network. Many fault tolerant routing algorithms are used to overcome the faults in Network on chip. However, these routing algorithms, suffer from another's problems like the congestion. In this work, a novel approach inspired by Catnap is proposed for NoCs using Local and Global congestion detection mechanisms with a hierarchical sub-networks architecture. With the help of these two techniques, the NoC becomes fault tolerant and is able to efficiently utilize the throughput. The simulation shows that the proposed algorithm gives a better performance by reducing the latency and increasing the reliability of the network. In addition, the algorithm has another advantage: it reduces the congestion which is considered as a temporary fault. Simulations show that our proposed algorithm reduces the latency more than 15% and throughput is improved by 20% compared to the PDA-FTR routing.

Keywords: *Network on Chip, Fault Tolerance, Congestion, Reliability, Sub-Network, Routing Algorithm*

1. Introduction

Network on Chip (NoC) has emerged as an efficient architecture to manage communication in a system on chip (SoC), where a large number of components and storage blocks are integrated on a single chip. This intensification of communications leads to important questions such as performance and energy consumption. Decreasing the transistor size has made semiconductors more sensitive to faults. Thus the challenge is, to maintain the system functionality during its operational lifetime and ensure that the system performance is preserved. For this reasons, researchers have attached a great deal of importance to the reliability in networks on chip. Faults which may occur in networks on chip can be divided into two main categories: permanent faults (or hard faults), and temporary faults (or soft faults) [1]. The soft faults are classified into transient and intermittent. These three types of faults (permanent, transient and intermittent) can be caused by several internal or external factors. The majority of failures (80%) are caused by transient faults, whilst the rest of them originate mainly in permanent and intermittent faults [2].

Received (April 15, 2018), Review Result (July 16, 2018), Accepted (July 26, 2018)

[*] Corresponding Author

Copyright ⓒ 2018 SERSC Australia

Faults in different components of the NoC have different causes, however, all can result in serious consequences such as loss of packet data, misrouting, deadlocks, and malfunctions. It follows that, the reliability of communication becomes an attracting challenge when designing the NoC. However, the communication has a huge impact on the performance of the network on chip, so it is very desirable to design efficient algorithms to ensure that. Faults in routers or links are the major problem that cause the failure of a packet transmission in a NoC. The communication performance of a NoC depends highly on the routing algorithm, which determines the path that each packet follows between the source and the destination node.

The fault tolerance routing algorithm is the process of finding a new and optimal path to steer packets from sources to destinations in a faulty network. Thus, the routing algorithm can efficiently increase the network performance. The congestion is another key factor which leads to increase the transmission delay and the power consumption. For this, these algorithms bring solutions for routing packets through the less congested regions and distributing the traffic over the network. Finally, failures and congestions should be managed in an effective way to ensure availability and robustness into the network on the chip.

In this context, many fault tolerance routing algorithms have been proposed for critical applications. It is, therefore, essential to handle failure, ensure correct and continuous operation of the circuit in its environment, even when the failure rate is high. We describe in this paper an interesting solution jointly to the congestion management and fault tolerance in NoC called DINRA -FTNoC. The new architecture offers many advantages like reducing latency and using alternative paths to route packets in case of faulty links or/and routers. The rest of the paper is organized as follows. Section 2 gives a brief overview of the work. The new architecture and implementation details of the proposed solution are presented in the third section. In Section 4, DINRA-FTNoC is evaluated. Our conclusions are drawn in the final section.

2. Related Work

Many solutions have been proposed to sustain the reliability of NoCs. These solutions can be classified based on the fault type (permanent or temporary), the type of application (critical or not critical) and finally the treatment mechanism (*e.g.*, using routing algorithms or architectural solutions). The architectural solutions include component redundancy, reconfiguration, and retransmission. Other solutions can combine the two techniques. But most of them focus only on one type of faults. Network congestion and faults are the main factors that degrade the NoCs performance. When a link or switch is faulty, the routing algorithm should choose another fault-free route for transferring the packets and avoiding the corruption of data. The main challenge to increase the reliability and provide a good performance is to deal with the principal problems in routing algorithms such as deadlock, congestion and failures. In the literature, there are many various approaches which have been suggested to solve this issue. Figure 1 summarizes some significant works about fault-tolerant techniques in NoC.

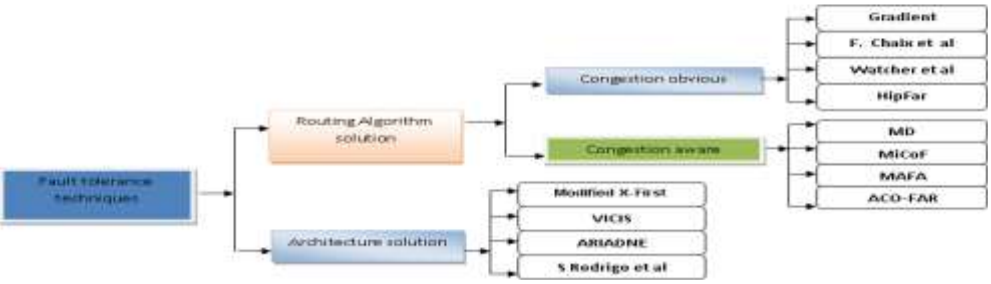


Figure 1. Some Related Works on Fault-Tolerant Techniques in NoC

Authors in [3] have developed a new routing algorithm which creates a path to route the packets through a cycle-free contour surrounding a faulty router. The algorithm uses 9 canonical contours of a mesh NoC to establish paths at runtime. Once a packet cannot find a route using the normal algorithm, due to faults in the system, the routing path is changed by connecting the different contours to establish a path. Specific turns in the path are achieved to avoid a deadlock. The major drawback of this algorithm is: it can tolerate only one faulty node.

Vicis [4] is a NoC design that can tolerate the loss of many network nodes/links and can also operate when only half of its routers are fully functional. A built-in self-test (BIST) circuit is used in each router to check hard faults in the system and to correct them by using ECC. This solution is costly and it fails to take into account the overhead.

Another work [5] proposed Explicit Path Routing (EPR). This routing algorithm tries to limit the latency degradation of packets even under faulty conditions. Depending on the path of the packet, some turns are prohibited to avoid the deadlock. An echo packet is sent when the system is reconfigured, after having established the optimal path of a source-destination pair. Although this approach is interesting, it suffers from the problem of the congestion management. Another main drawback is that it utilizes the virtual channels which are expensive in term of resources.

The Gradient [6] is an adaptive routing algorithm to tolerate faulty node. The algorithm divides the entire system into eight zones by gradient line. Each zone is provided with one main path and two alternative paths for the packet delivery when the main routing path is faulty. The main disadvantage of these approaches is that they do not foresee an efficient mechanism to control the congestion problem, in order to make a good decision routing for complex traffic condition. To avoid congestion and failure at the same time, the solution proposed by [7] and [8] adopts a minimal path to reduce the congestion caused by the presence of faults. These algorithms are able to select the shortest path to route packets as long as a path exists. In [9] authors proposed another routing algorithm which is inspired from ACO called ACO-FAR to perform a load balancing with low latency and a high throughput. The three algorithms mentioned above use minimal path, therefore they are all live lock free.

For maintaining a good performance in the network, the work in [10] present a simpler method called MiCoF. When a faulty router is detected, packets take an alternative minimal path already foreseen in the architecture. This algorithm selects the shortest path.

To increase the reliability and the performance of NoCs, the authors of [21] have proposed a new routing algorithm called MUGEN. This work comprises an optimized method to exchange messages between different virtual channel classes. The selection function uses distant router link information to avoid deadlock and a new congestion metric used to guide routing decisions towards less congested areas. The authors claim that the proposed algorithm present promising results in terms of fault-tolerance and performance.

A recent work proposes an algorithm called PDA–FTR [22] (Path-Diversity-Aware Fault-Tolerant Routing) which considers simultaneously the path diversity and the buffer information to achieve a fault tolerance and a load-balancing. The drawback of the proposed fault-tolerant routing algorithm is that the information used provides a limited view of traffic in the network, which can engender a congestion in heavy traffic.

Copyright ⓒ 2018 SERSC Australia

Table 1. Most Known Fault-Tolerant Routing Algorithms

FT Routing Algorithm	Years	Components	Type	Congestion
[3]	2008	Up One switch failure	Permanent	No
[5]	2010	Many faults switch	Permanent	No
[11]	2011	Links and switches	Permanent	No
[12]	2012	Links and switches failures	Permanent	No
[13]	2012	Links and switches failures	Transient faults	No
[14]	2013	Links	Permanent	No
[15]	2013	Up to 6 faulty links	Permanent	Yes
[6]	2013	Switch failures	Permanent	No
[8]	2013	Up to 6 faulty nodes	Permanent	Yes
[10]	2013	Up to 2 faults Link	Permanent	Yes
[9]	2013	Any number of faults links	Permanent	No
[17]	2013	Switch	Permanent	No
[16]	2014	Up to 4 faulty switches	Permanent	Yes
[18]	2014	Many faulty switches	Permanent	No
[19]	2014	Links and switches	Permanent and transient	Yes
[20]	2014	All faulty links/nodes	Permanent	No
[21]	2016	Links and switches	Permanent	Yes
[22]	2017	Switches	Permanent	Yes

Table 1 recaps the most popular fault tolerance routing algorithms. The all approaches mentioned above have the following benefits and weaknesses: (a) under high traffic conditions and (b) the performance in term of throughput and latency degrade in case of important rate of faults in links and routers. Finally, all of them deal with permanent faults. Hence the aim is to achieve a robust routing algorithm with these required functions:

(a) the ability to avoid the congested nodes for having a balanced traffic,

(b) tolerate high rate of faults which may affect links and routers and,

(c) tolerate transient error which appears in communication process.

All the techniques described above bring their approaches to the fault-tolerance problem, but they have a cost in terms of performance to be considered in the area of routing algorithms design. This is evaluated by a set of metrics, such as: latency, throughput, network congestion and energy consumption. To satisfy all these requirements at the same time is impossible. Thus, the challenge for the researchers is to find a good compromise between these costs and the reliability.

Copyright ⓒ 2018 SERSC Australia

3. Network Architecture

Networks-on-chip have been a very active research field since their appearance in early 2000s. Since then, many architectures have been proposed in the literature. Routers, Network Interface (NI), IPs and links are the main elements in a NoC. In this section, we introduce the proposed architecture and its main components. Before giving the details, it is necessary to specify that we have adopted two important assumptions: First, the links connecting the Intellectual Property (IP) to the input and output Network Interface (NI) are always non-faulty. Second, we assume that there is at least one path between a (source, destination) pair. These hypotheses are necessary to deliver any packet from source to destination.

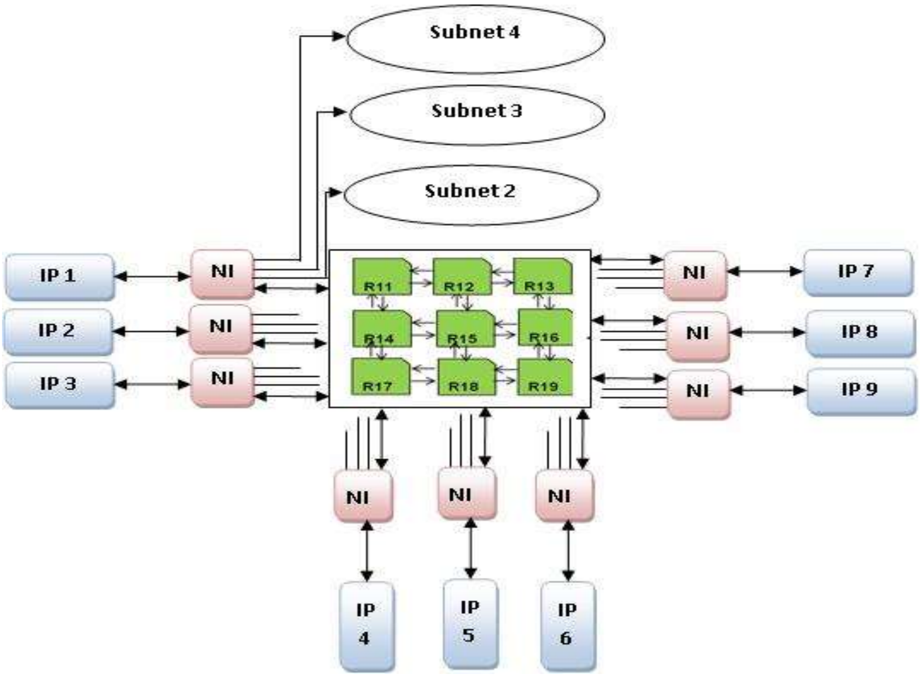


Figure 2. The Global Architecture
NI: Network Interface, R: Router, IP: Intellectual Property

Figure 2 shows the global architecture. The network on chip is divided into subnets. Every IP is connected to four routers, each one belonging to a disjoint subnet. Thus, every switch belongs only to a single subnet. For example, we build 4 subnets with 6*6 switches, each one can communicate only with the others of the same subnet. This method represents an innovative alternative to get an efficient reliability. When a subnet is congested or faulty, we should disable the entire subnet and isolate it. When focusing on subnets, we can see that they have the same connectivity pattern.

The three IPs located in the east of the NoC are connected to the three switches of the same subnet. The same topology is applied for the west and the south IPs. We have implemented an algorithm (Section 3.5) to select the subnet. We have added 2 bits to the header flit in NI for distinguishing each subnet from the others. The key advantage of this architecture is that the connectivity pattern has several alternative paths that can be used for increasing the fault tolerance, as it will be shown in Section 3.5. When the first subnet is inoperable, we activate the second subnet, and when the second subnet is defective, we active the third subnet, and so on. The worst case scenario is when the top-subnet is faulty or congested. To provide a higher fault-tolerance, the new architecture requires an additional links. We solve this by a short connection between end nodes and IPs.

3.1. Router Architecture

A Router is an entity that facilitates communication between IP cores in Networks on chip. We introduce in this part the proposed router architecture and its main components. The router is the back-bone component of the DINRA-NoC design. The two necessary components which are added to the basic router architecture are shown in Figure 6. Each end node has a maximum number of 4 input and 4 output ports. One input/output port is used to connect the switch to the NI. The number of input-ports depends on the router position in the subnet with the perspective to eliminate the extra ports and reduce the area overhead and the power consumption. Each router can be connected to a maximum of four adjacent routers as well as the local intellectual property (IP) across NI.

Figure 2 illustrates the NoC topology which is a 6 x 6 sizes using a wormhole switching policy and the credit-based flow control. To locate and to distinguish between routers in the sub-network, we give every router a unique address defined in XY coordinates. To identify each router, we define two parameters:

- SN (ID): Sub-network identification. It is a number indicating the subnet,
- (X, Y): Denote the coordinates of the router in its SN.

The routing unit considers these three (X, Y, SN ID) addresses to transmit a packet. Indeed, the SN ID is a binary number defined by 2 bits, and each sub-network has its unique ID code (see Table 2).

Table 2. Coding of The Different Subnet

ID SN	Value
00	Subnet 0
01	Subnet 1
10	Subnet 2
11	Subnet 3

3.2. The Proposed Routing Algorithm

This section focuses on the routing algorithm proposed in this paper. To deal with all the requirements of the fault tolerance, in the initial stage of the process, the proposed solution can provide an online detection of permanent and transient faults. Once the detection has been done, we can do the isolation of the faulty components. As soon as these steps have been carried out, the routing algorithm ensures the delivery of packets to their destination when a path exists. At this step, we can indicate if the destination is unreachable or not. In addition, routers do not require any virtual channels. They work in a fully distributed way to transmit the packets in case of faulty nodes. In the proposed routing algorithm, the path of the packets depends on the congestion status and/or faults in the network. When the sub-network is congested or faulty, another subnet is chosen.

3.3. Congestion Detection

Wormhole routing requires less memory than the virtual cut through or store and forward routing strategies. This may cause a congestion state because the buffer requirements may vary based on the application. The increasing number of cores will contribute to an additional traffic which will increase the congestion as well. So, there is a need to decrease the congestion and enhance the performance at the same time for greater size NoCs. When the traffic raises or exceeds a determined level, the latency increases and consequently the throughput decreases. Inspired by Catnap [23], our routing selection is made depending on the subnet's congestion. This congestion information is obtained by the stop signal issued from the RCS (Regional Congestion Status) used in our NoC system. For the congestion detection in each subnet, we have used only a single additional bit in the buffer of each node, which is the minimum extra required for the detection.

3.3.1. Local Congestion Detection

The NI of a node can measure the injection rate, which is defined by the number of flits injected into the network over a period of time (4 cycles). So, if the router BFM (maximum buffer occupancy) is greater than a certain threshold, then this subnet is considered to be congested, and a local congestion status (LCS) bit is set to true. The BFM congestion detection mechanism is local to a network node, according to Catnap. The best performing thresholds for various regional congestion detection policies is BFM: 9 flits. This information is stored in the Network interface. Table 3 shows the codes associated to the different SNs states. So, to measure BFM, the NI connected to each router keeps track of the buffer occupancy of each router input buffer.

Table 3. Coding of The Different States for Each Subnet

State SN	Code
Normal	00
Congested	01
Out of order	10
Disable	11

3.3.2. Regional Congestion Detection

We use a simple 1-bit OR network which is set to true, when any of the routers LCS is true. This bit value, to which we refer as the regional congestion status (RCS), can be read and communicated to all the nodes in the same subnet. The OR-network is architected as an H-Tree network. The NI of a node sets its RCS if the local congestion status (LCS) is true. LCS is determined and based on the BFM of the local router. A node NI detects the congestion for a subnet if either the local congestion status (LCS) is true (based on BFM of the local router), or if the regional congestion status (RCS) is true.

3.3.3. Subnet State

When a packet reaches the head of NI, one of the subnet is selected (based on state of subnet), and the packet is injected into that subnet see Table 3 (First subnet by default). All the flits of a packet travel in the same subnet until they reach the destination. The NI is connected to several routers as shown in Figure 2, each router belongs to a different subnet.

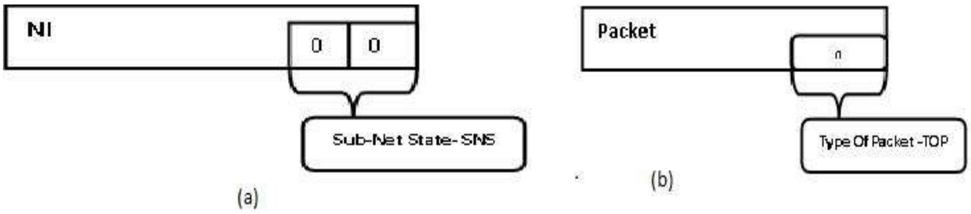


Figure 3. (a) Subnet State, (b) Type of Packet (ToP)

We distinguish two types of packets: critical and non-critical packets. We use 1 bit for distinguishing ToP (Type of Packet) as shown in Figure 3 (b). This information is added

Copyright ⓒ 2018 SERSC Australia

to Header Flit of each packet. So, ToP equals 0 if the packet is non-critical. Otherwise its value is 1.

3.4. Fault Detection

For on-chip networks to handle the faults in routers, links and network interfaces, BIST [24] is the traditional solution. This method is a fault detection scheme. But this technique requires external circuits to perform error detection, and time to test who can reduce the performance of the NoC in term of latency. To develop our approach, we have opted for the Cyclic Redundancy Cycle (CRC), a solution that can be used to perform error detection by comparing the input and output of each router to detect wrong packets and faulty components in a NoC. This approach is inspired by [24], with some modification. Each router is equipped with a Test Module (TM). The goal of this mechanism is to enable online fault detection. The CRC polynomial used is g = 1 + x + x4, so we add 4 bits to the header of each packet. When an error is detected anywhere in the router/link, we know that the error exists but we can’t correct it with CRC decoding.

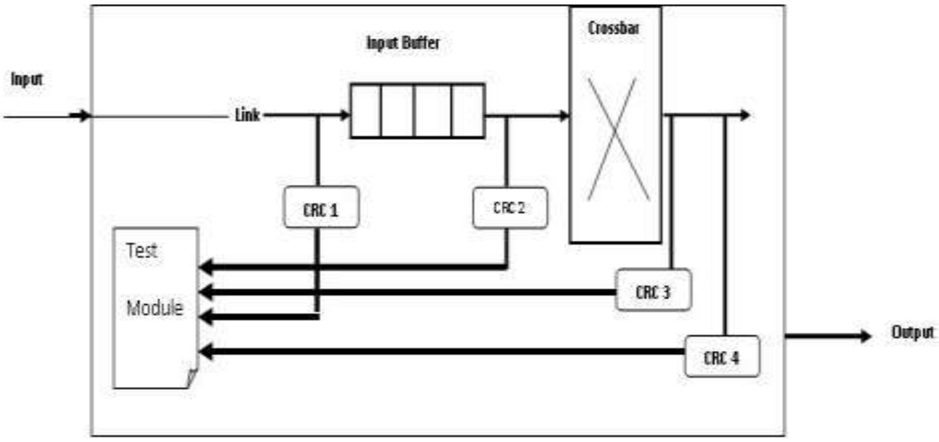


Figure 4. Internal Architecture of the Router

To test and diagnose the communication infrastructure in the NoC, first of all, it is necessary to detect the fault. Secondly, there is a need to identify the faulty components: routers, links or cores. Finally, we analyze how to use efficiently the remaining components that are fault free. In the router, the fault can occur in all the components (crossbar, buffers and others). At this high core density in networks on chip, considering faults only in the routers or links do not provide the optimal safety. Other components such as input-buffers and crossbar should be given prominent attention to ensure the fault tolerance and improve the system reliability.

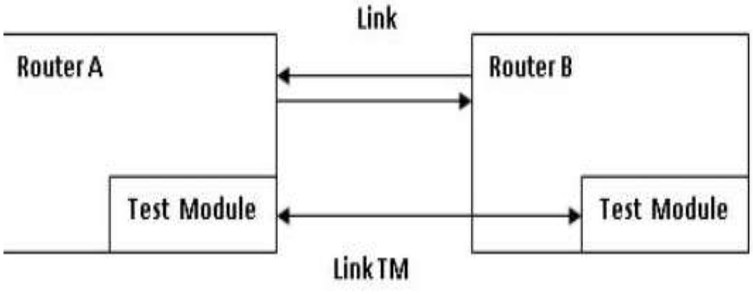


Figure 5. Communication Between Two Neighboring Routers

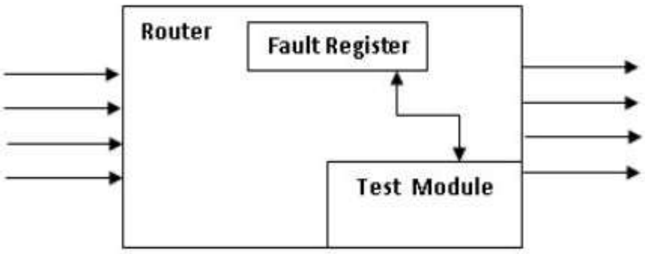


Figure 6. Communication Between the Fault Register and the Test Module

To achieve this goal, we present in this section the Test Module (TM) added in our router (see Figure 5). The TM is one of the main components of our system, because it manages the diagnosis from all kinds of faults in four main components: inter-router links, input-buffers, crossbar, and header flit. We start in this subsection to describe the main functionalities of the Test -Module (TM) and its important role in orchestrating the different process inside the router. Test Module detects and locates the faulty routers or links. Therefore, in the same subnet, only adjacent routers can communicate this information.

This information is always sent to the FR (Fault Register), in order to keep the fault occurrence in the input-buffers, the crossbar and header flits. The aim is to use this information during the selection of the next port. We have used a simple fault detection mechanism based on a Cyclic-Redundancy-Check (CRC) in each output component of the router that reads the incoming flit to detect any error. Depending on this verification, the Test-Module (TM) sends a single-bit signal to the TM of upstream node that can be either 0 or 1, for respectively valid or faulty, as shown in Figure 5. Each router sends the collected information corresponding to its own fault status to all the neighboring nodes. This information is represented in a 3-bit signal (see Table 4) representing the router/link status in each direction (North, East, South, and West). Depending on these states, our algorithm (DINRA) reads the fault status of the next nodes/links received from the TM and checks the number of possible safety directions. For example, when a fault is detected in the buffer, a signal is sent to inform the TM module about the fault presence. The same case is applied for the other components.

When receiving this signal, the TM disables the entire output-port and the faulty router to save the dynamic power. At the same time, the TM updates the FR-status array by flagging the link connected to the faulty router. This information is constantly sent to all the TMs for all neighboring nodes.

Finally, to keep the faults information in routers or links, the TM interacts with the FR unit to exchange fault information and control signals, as shown in Figure 6.

Table 4. Codification of Different States of Links and Routers

State	Value	Description
1	00	Port East unsafe
1	01	Port North unsafe
1	10	Port South unsafe
1	11	Fail Router
0	00	Port East safe
0	01	Port North safe
0	10	Port South safe
0	11	Safe Router

Copyright ⓒ 2018 SERSC Australia

3.4.1. Permanent Fault Detection

The permanent errors cannot be eliminated by a simple reset of the circuit. The routers adjacent to a router with a permanent fault are informed about its state. This is done to prevent any traffic to this defective router (for example, by disabling the output ports leading to this router). The same case is applied for defective links.

3.4.2. Transient Fault Detection

The transient errors occur typically during a very short time and are not destructive. So, after k attempts, the test module can consider the (temporary) dynamic faults as permanent. So we want to tolerate several on-line faults without resetting the circuit.

3.5. Routing Algorithm

The packet routing is one of the significant factors in the design of NoC architecture. To achieve the fault tolerance, the routing algorithm needs multiple paths to route the packets for each pair of source destination. First, the proposed routing algorithm must be adaptive to choose between them. Second, the topology must provide an alternative route. Complex routing algorithms can introduce extra area and energy overhead. In our proposed solution, the main goal is to maximize the performance in terms of latency and reliability by having a good successful packet delivery rate.

Algorithm: path computation

```
State_sub_s0 = Active & RCS =0;
For (s=0, s = 3, s++) /*– Source S (x, y, z), Destination D (x’, y’, z’)
If state_subnet_s = 00 /*– — normal state 4 If RCS = 0 then
For each S, D
North Last /*———-Routing Algorithm by default
if (link state && router) == unsafe
South last /——-Routing Algorithm
else / —– Activate s+1

State_sub = Faulty; s=s+1;
Else s+1 / ——— Next Subnet
State_sub = Congested; s=s+1;
state_sub_s= Active /——active next subnet
End loop
End loop
```

Copyright ⓒ 2018 SERSC Australia

Here are some explanations about our algorithm: NI connected to a router first inject a packets into the subnet-0 (line 2), and all the flits of a packet travel in the same subnet until they reach the destination. If the subnet-0 is congested or faulty, a higher subnet (s+1) can be activated more quickly in time to avoid a performance loss (line 11) and same case is applied for others subnets. To route packets in same subnet, we use the Last North routing algorithm, and if can't, we use the South Last to avoid faulty components (Links or Routers-line 8). And, if can't route the packet in second case, we active the higher subnet (s+1) and we set state-subnet to faulty. The same process is settled if the current subnet is it going congested-line 12.

3.5.1 Fault-Tolerance

DINRA has all information about the fault status of next links and nodes. It has three possible directions for routing the flits. The congestion status is the higher priority. We adopted this condition to ensure the fault tolerance and a better performance either in the presence or the absence of faults. When there is no valid route available, DINRA chooses another route in the second subnet priorities (path diversity and congestion) as illustrated in the algorithm.

Wormhole-switching requires less memory than the virtual cut through or store and forward routing strategies. In this way, DINRA adopts a Wormhole-switching and the packet forwarding can be executed in an efficient way for maintaining a small buffer size.

3.5.2. Deadlock

This algorithm combines two adaptive routing algorithms North-Last and South-Last. Both of them use restrictions to avoid deadlocks [25]. The deadlock problem may arise with the adaptive routing. Most of them use Virtual-channels (VCs) or Turn Model to the routing selection to avoid the deadlock.

3.5.3. Adaptability

One of the key properties of the DINRA architecture is its capability to be adapted for providing several routes to transfer packets for each source destination node. This increases significantly the reliability of the entire NoC. DINRA proved its yield by ensuring both fault tolerance and congestion aware.

3.5.4. Retransmission of Packets

There exists a case where some of given packets are stored in the buffer and the output link or the router becomes suddenly defective during the packets transit. This modifies the validity of the path (dynamic fault), which later will be proved as faulty. This corruption creates sub-packets which cannot arrive to their destination. A copy of the Header of this packet will be stored in a special buffer. A notification message is sent to the source node to transmit all the packets again. Also, we must remove the flit's which have been transmitted in the case of a non-reachable destination or only to transmit the packets not yet transmitted in the case of a temporary fault. See Figure 7.

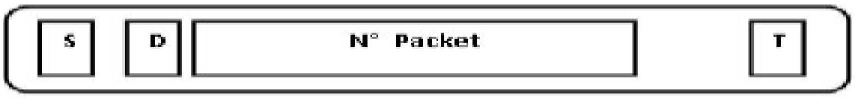


Figure 7. The Notification Message Format
S: Source, D: Destination, T: Type of Packets

According to the network status, there are three cases as follows: Case I, when the packet is being sent from the current IP to another and the current subnet is congested.

Case II and III, when the packet is being sent from current router to the next hop, in this case, the destination IP can be reachable or not.

There are two cases where a packet is considered as lost and must be retransmitted another time by the source (Case II and III):

- A packet fragment arrives in a faulty subnet: for example, a router without healthy output or a router/packet is blocked by routing restrictions to its reachable destination.
- The second case: the destination is not attainable when a packet fragment cannot reach the destination node before a" Timeout" T (an empirical value that varies the depending on the size of NoC).

In order to handle and tolerate temporary faults which can occur during the transfer of packets (For example, when a router/link becomes suddenly defective), a notification message is sent to the source node to retransmit again all the packets which have not reached their destination.

4. Evaluation

We devote this section to assess and analyze the performance of the proposed system. We have selected two traffic patterns: Random and Shuffle traffic to evaluate the performance of the proposed system. The first sets of analysis investigated are Latency and the Reliability of the proposed routing under each of the aforementioned patterns traffic. We observed the performance variation of DINRA under different fault-rates of links and routers (0%, 5%, 10%, 20% and 40%). During the evaluation, we divided the faults into two parts: the biggest parts are allocated for transient faults and the remaining portion is considering for permanent. To evaluate our algorithm, we use a public available simulator: Noxim.

The Noxim simulator is developed using System C and provides a command line interface for defining several parameters of a NoC. In particular, the user can customize the network size, buffer size, packet size distribution, routing algorithm, selection strategy, packet injection rate, traffic time distribution, traffic pattern when use and hot-spot traffic distribution. The simulator allows NoC evaluation in terms of throughput, delay, and power consumption. Such information is delivered to the user both in terms of average and per-communication results. In detail, the user is allowed to collect different evaluation metrics, including the total number of received packets/flits, global average throughput, max/min global delay, total energy consumption, per-communications delay/throughput/energy and others.

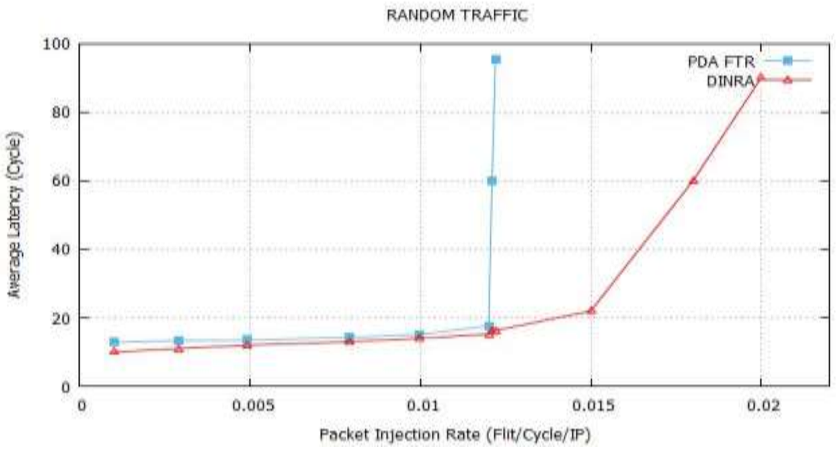


Figure 8. The Average Latency of DINRA and PDA-FTR in 8*8 with Random Traffic, in Four Faulty Routers

Copyright ⓒ 2018 SERSC Australia

4.1. Latency Evaluation

In this party, to evaluate our approach, we have varied the rate of injection of faults and we have measured the average latency. First, we have evaluated the communication latency of DINRA by calculating the average latency/flit. Figure 8 illustrates the latency/flit results under Random traffic pattern. When no faults are detected (0%), these tests revealed that our solution reduce the latency with an average of 19.4% compared to PDA-FTR routing algorithm. As we stated previously, DINRA takes advantage of the adaptive routing to forward the flits to the next neighboring node which is faster than PDA-FTR routing algorithm. Moreover, DINRA takes into consideration the congestion status and the traffic balance (see line 8 algorithm). As expected, for high values of fault-rate, DINRA performs better than PDA-FTR. As a result, the latency/flit along the network is reduced. When observing the latency variation over different fault-rates, DINRA performs better than PDA-FTR even under 10% fault-rates, and when this rate reaches the 25% and 40%, the latency increased with only 12% and 30% respectively, see figure 10 and 11.

As expected, faults injected affect the average latency of the network. This one tends to increase with the increase of the network load and also with the increase in the number of router/link faults. It can be seen in Figure 8, 9 that the average latency of DINRA increases if we inject more traffic and / or more faults.

This performance is strongly related to the new architecture, whichever the choice path is more diverse. In this manner, when a faulty link is detected in one path, there still exist other ones with valid links in the same subnet, otherwise in others. DINRA takes advantage of this property to route the flits. Thus, the congestion is more important, because more flits will be blocked in the buffers. This explains why the performance drops at higher fault-rates. This is mainly caused by the traffic congestion. To better see the effects of DINRA routing, we observe the latency results of the Random traffic pattern illustrated in Figure 10. In the case where no faults are observed, DINRA still reduces the latency with 16.29% compared to PDA-FTR. The latency gets its highest value at 45% fault-rate with a high flit injection (100.000 flits) and reaches the 50% (more than PDA-FTR results).

In addition to the congestion caused by the increasing faulty links and routers explained above, we have also an important impact on the latency.

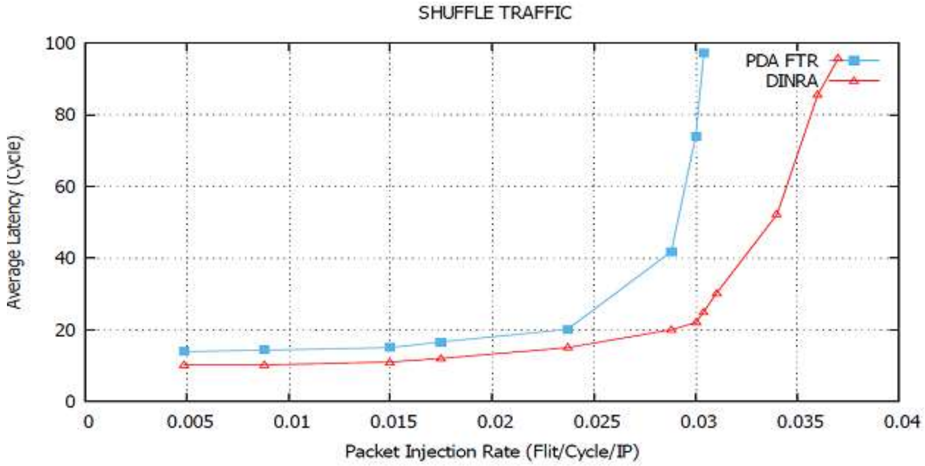
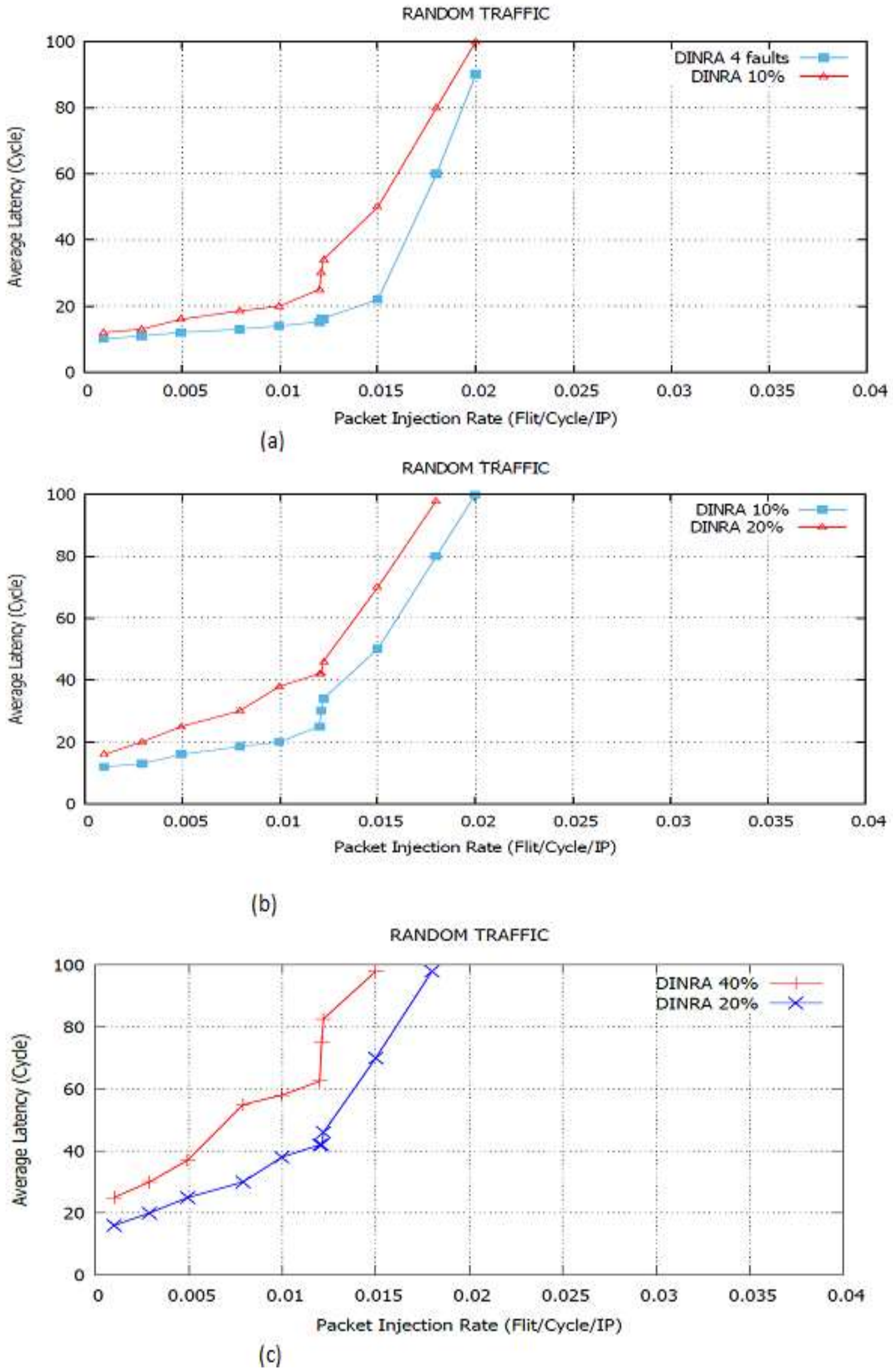


Figure 9. The Average Latency of DINRA and PDA-FTR with 8*8 Under Shuffle Traffic, in Four Faulty Routers



RANDOM TRAFFIC

(a)

RANDOM TRAFFIC

(b)

RANDOM TRAFFIC

(c)

Figure 10. Average Latency of DINRA with Different (PIR) Package Injection Rates "Random Traffic" for a 12x12 NoC (a) 10%, (b) 20%, (c) 40% Faults

Copyright ⓒ 2018 SERSC Australia

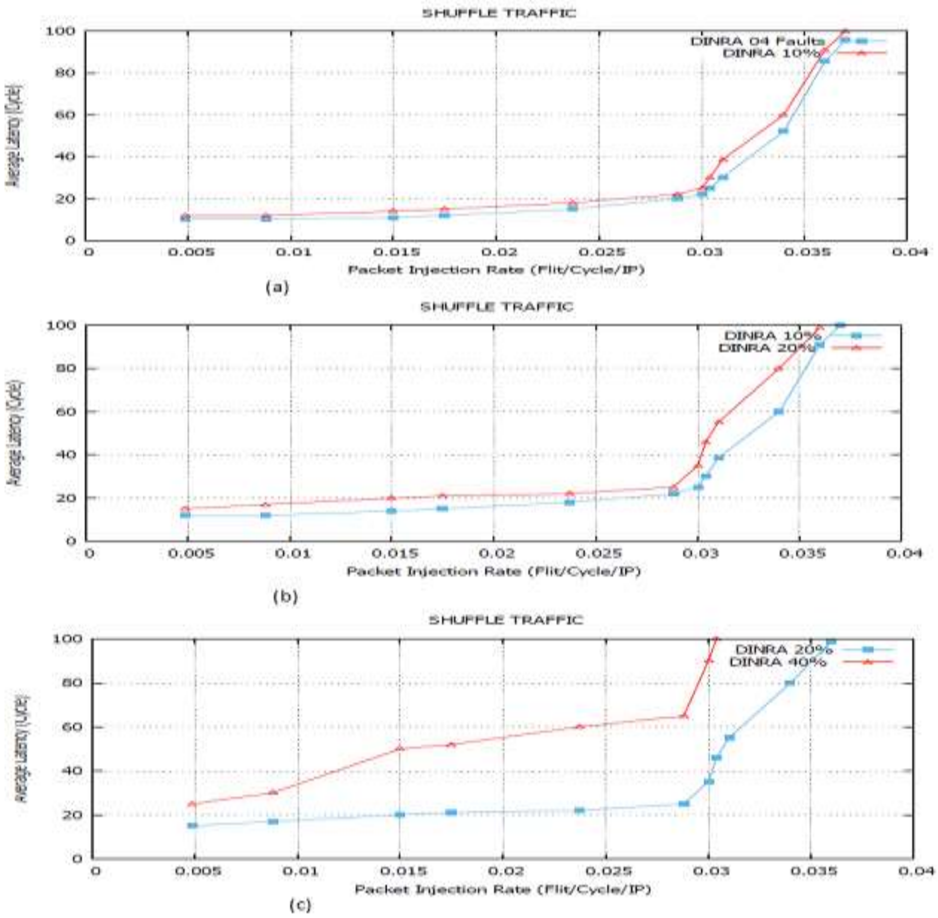


Figure 11. Average Latency of DINRA with Different (PIR) Package Injection Rates “Shuffle Traffic” for a 12x12 NoC (a) 10%, (b) 20% and (c) 40% Faults

4.2. Reliability Evaluation

In this subsection, we discuss the reliability of DINRA. We recall that the reliability is the capability of a system to deliver all the packets to their destinations. For the second evaluation, we have calculated the success rates of each algorithm using the two patterns traffic (equation 1). We can explain this reliability by the fact that DINRA always finds a path from a source to a destination, no matter where the fault is located and how heavy the traffic is. The only possible problem that can affect the arrival of a given packet to its destination is the presence of a deadlock. However, thanks to the hybrid turn model adopted, deadlock is avoided. Therefore, the most striking result to emerge from the data is that all packets can reach their destinations which provide a high reliability under different fault-rates and injections.

We have done the same series of experiments but this time taking the model of router faults (Figure 13). Again, we have seen an increase in average latency based on the number of faults and the packet injection rate. It should be noted that the average latency increases here faster than in the previous case when only the links were considered defective. A faulty router is considered a router with its entire faulty links. In case of faulty routers, for DINRA algorithm, there are at least for each packet one path founded to its destination. Nevertheless, with more than 30% of the routers failing, the service is still rendered but with a lower packet injection rate depending on the type of traffic, compared to a fault-free situation.

Copyright ⓒ 2018 SERSC Australia

$$\text{Successratio} = \frac{\text{Total. arrived packets}}{\text{Total injected packets}} \times 100 \qquad (1)$$

Table 5. Comparison of Success Ratio % with PDA-FAR Routing Algorithm

No of faults	Gradient	PDA-FTR	DINRA
2	2.8%	0.07%	00
4	3.2%	0.5%	00

We see that DINRA makes it possible to guarantee a high level of reliability. In the case of a 12x12 NoC size, it can deliver more than 95% of messages when 10% of links are defective. When we have 40% of the links defective, the delivery rate successfully varies from 50 and 60% for" Random Traffic" (Figure 12). When considering node faults (router), the rate of packets delivered successfully decreases significantly, and this show the great impact in reliability of a faulty router. Nevertheless, even in the scenario where 40% of routers are defective, the delivery rate varies from 0 and 10% reaches 10% for "Uniform Random" traffic (Figure 13). With the addition of packet forwarding and congestion-aware techniques, DINRA is clearly improving performance and reliability over PDA FTRs. fault injection affects average network Latency and Throughput, which increases with the number of link faults, routers and the number of packets injected into the network.

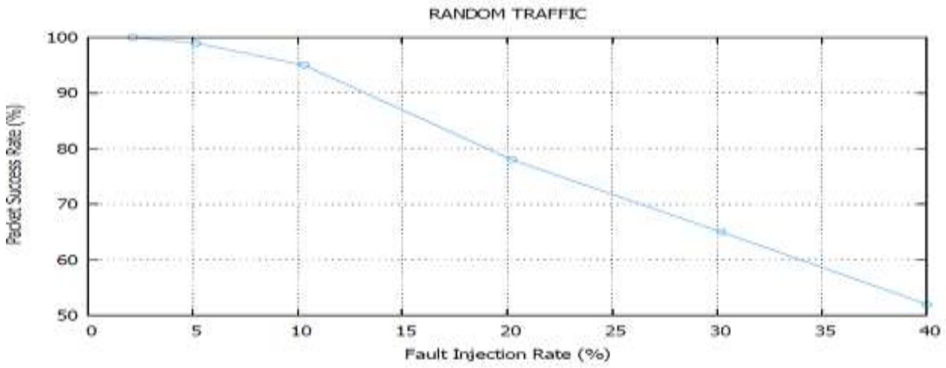


Figure 12. Rate of Successfully Delivered Packets with Different Fault Injection Ratios (Link Fault) Under "Random" Traffic for A 12x12 Noc

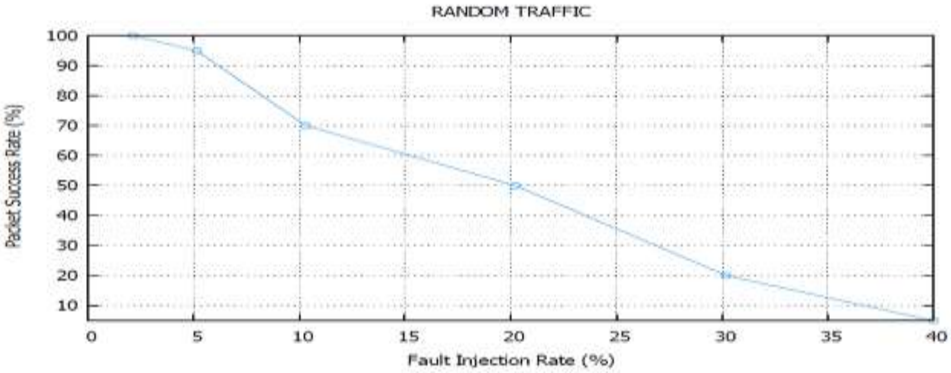


Figure 13. Rate of Successfully Delivered Packets with Different Fault Injection Ratios (Router Fault) Under "Random" Traffic for A 12x12 NoC

Copyright ⓒ 2018 SERSC Australia

4.3. Throughput

We have evaluated the throughput, which is defined as the accepted traffic of the network at a given latency. The throughput of network is presented by the formula below:

$$\text{NetworkThroughput} = \text{Saturation Throughput} \times \text{No: of nodes} \qquad (2)$$

We then wanted to determine the impact of the Throughput and the size of the network. For this, we have measured the Throughput for four NoC sizes and for one single failure situation. (Figure 14) summarizes the simulations performed for various sizes. We can see, as expected, that our experiments confirm that DINRA is suitable for both small and large NoC systems.

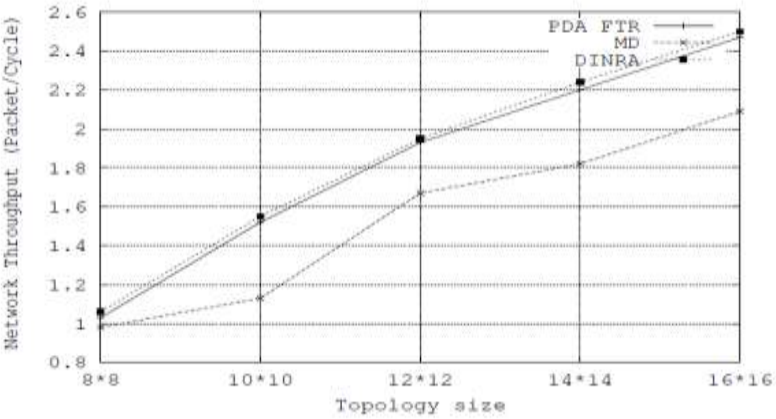


Figure 14. Throughput of Dinra Under Different Topology Sizes and in Single Fault

5. Conclusion

In this paper, we have presented an innovative fault- tolerant routing algorithm, called DINRA, for the Networks-on-Chip (NoC). The algorithm operates with a proposed architecture which aims to avoid a heavy sys-tem failure when some faulty components (routers and links) are observed in a NoC. The proposed solution preserves the network performance. The evaluation done proves that DINRA performs better than PDA-FTR algorithm, even at 40% fault-rate in links and routers. Despite the good results obtained, our work clearly has some limitations which should be fixed to enhance its performance and reliability. The first one is that we need to analyze the best techniques which can be implemented for fault-detection. Second, it is also interesting to study if the time required to handle a fault-detection can be improved. We are currently studying the power consumption and hardware complexity of the proposed architecture. In a future work, we plan to focus on these parameters.

References

[1] M. Radetzki, C. Feng, X. Zhao and A. Jantsch "Methods for fault tolerance in networks-on-chip", ACM Computing Surveys, **(2013)**, pp. 1-36.

[2] T. Lehtonen, P. Liljeberg and J. Plosila, "Online reconfigurable self-timed links for fault tolerant NoC", VLSI Design, **(2007)**, pp. 13.

[3] Z. Zhang, A. Greiner and S. Taktak, "A reconFigurable routing algorithm for a fault-tolerant 2D-Mesh Network-on-Chip", Design Automation Conference, DAC 2008. 45th ACM/IEEE, **(2008)** June, pp. 441-446.

[4] D. Fick, A. DeOrio, J. Hu, V. Bertacco, D. Blaauw and D. Sylvester, "Vicis: a reliable network for unreliable silicon", ACM/IEEE Design Automation Conference (DAC), **(2009)**.

[5] F. Chaix, D. Avresky, N. E. Zergainoh and M. Nicolaidis, “Fault-tolerant deadlock-free adaptive routing for any set of link and node failures in multi-cores systems”, In Proceedings 9th IEEE International Symposium on Network Computing and Applications, NCA, **(2010)**, pp. 52-59.

[6] I. Pratomo and S. Pillement, “Gradient an adaptive fault-tolerant routing algorithm for 2D mesh network-on-chips”, Proc. Des. Archit. Signal Image Process, **(2012)** October, pp. 1-8.

[7] M. Ebrahimi, M. Daneshtalab, J. Plosila and H. Tenhunen, “MAFA: Adaptive Fault-Tolerant Routing Algorithm for Networks-on-Chip”, In Proc. of 15th Euromicro Conference on Digital System Design, **(2012)** September, pp. 201-207.

[8] M. Ebrahimi, M. Daneshtalab, J. Plosila and F. Mehdipour, “MD: Minimal path-based Fault-Tolerant Routing in On-Chip Networks”, In Proc. of the 18th Asia and South Pacific Design Automation Conference, **(2013)** January, pp. 35-40.

[9] C.-A. Lin, H.-K. Hsin, E.-J. Chang and A.-Y. Wu, “ACO-based fault-aware routing algorithm for network-on-chip systems”, Signal Processing Systems (SiPS), Workshop on IEEE, **(2013)**, pp. 342-347.

[10] M. Ebrahimi, M. Daneshtalab, J. Plosila and H. Tenhunen, “Minimal- Path Fault-Tolerant Approach Using Connection Retaining Structure in Networks-on-Chip”, Proceedings of 7th ACM/IEEE International Symposium on Networks-onChip (NOCS), US, **(2013)** April, pp. 1-8.

[11] A. DeOrio, L.-S. Peh and V. Bertacco, “ARIADNE: Agnostic reconfiguration in a disconnected network environment”, International Conference on Parallel Architectures and Compilation Techniques (PACT), **(2011)**, pp. 298-309.

[12] A. DeOrio, D. Fick, V. Bertacco, D. Sylvester, D. Blaauw, J. Hu and G. Chen, “A reliable routing architecture and algorithm for NoCs”, IEEE Trans. Comput. Aided Design Integrated Circuits Syst., vol. 31, no. 5, **(2012)**, pp. 726-739.

[13] S. Xiangming, “Configurable redundant routing for network on chip”, IEEE, **(2012)**, pp. 477-479.

[14] A. Vitkovskiy, V. Soteriou and C. Nicopoulos, “Dynamic fault-tolerant routing algorithm for networks-on-chip based on localised detouring paths”, Computers & Digital Techniques IET, vol. 7, no. 2, **(2013)** March, pp. 93-103.

[15] M. Ebrahimi, M. Daneshtalab and J. Plosila, “High performance fault- tolerant routing algorithm for NoC-based many-core systems”, IEEE, **(2013)**, pp. 462-469.

[16] C. Chen and S. D. Cotofana, “An effective routing algorithm to avoid unnecessary link abandon in 2D mesh NoCs”, IEEE, **(2013)**, pp. 311-318.

[17] E. Wachter, A. Erichsen, A. Amory and F. G. Moraes, “Topology Agnostic Fault- Tolerant NoC Routing Method”, DATE, **(2013)**, pp. 1595-1600.

[18] B. Fu, Y. Han, H. Li and X. Li, “Zone Defense: a fault-tolerant routing for 2-D meshes without virtual channels”, IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 22, no. 1, **(2014)**, pp. 113-126.

[19] M. Dimopoulos, Y. Gang, L. Anghel, M. Benabdenbi, N. Zergainoh and M. Nicolaidis, “Fault-tolerant adaptive routing under an unconstrained set of node and link failures for manycore systems-on-chip”, Microprocessors Microsystems, vol. 38, no. 6, **(2014)** August, pp. 620-635.

[20] R. J. Behrouz, M. Modarressi and H. Sarbazi-Azad, “Fault-tolerant routing algorithms in networks on-chip, in: Routing Algorithms in Networks- on Chip”, Springer, **(2014)**, pp. 193-210.

[21] A. Charif, N. E. Zergainoh and M. Nicolaidis, “MUGEN: A high-performance fault-tolerant routing algorithm for unreliable Networks-on-Chip”, IOLTS, **(2015)**, pp. 71-76.

[22] Y. Y. Chen, E. J. Chang and H. K. Hsin “Path-Diversity-Aware Fault-Tolerant Routing Algorithm for Network-on-Chip”, Systems IEEE Transactions on parallel and distributed systems, vol. 28, no. 3, **(2017)** March, pp. 838-849.

[23] R. Das, S. Narayanasamy, S. K. Satpathy and R. Dreslinski, “Catnap: Energy Proportional Multiple Network-on-Chip”, In proceedings of the 40th International Symposium on Computer Architecture, Tel Aviv, Israel ISCA, **(2013)**.

[24] V. Fochi, E. Wachter, A. Erichsen, A. M. Amory and F. G. Moraes, “An Integrated Method for Implementing Online Fault Detection in NoC Based MPSoCs”, In: ISCAS, **(2015)**, pp. 1562-1565.

[25] C. Glass and L. Ni, “The turn model for adaptive routing”, Proceedings of the 19th annual international symposium on Computer architecture (ISCA 92), New York, NY, USA, **(1992)**, pp. 278-287.

Author

Nehnouh Chakib, obtained my Engineer degree (2001) from Ecole Nationale Suprieure d’Informatique (www.esi.dz) and my Magister degree (2010) from University of Mascara. Currently, I am an associate professor and researcher at the University of Chlef-Algeria. I am teaching, among others, Artificial Intelligence, Data Mining for the Master ISIA Class. My research interests lie in Swarm intelligence and Fault tolerance in Network on chip.

Copyright ⓒ 2018 SERSC Australia