Collaborative Routing Algorithm for Fault Tolerance in Network on Chip CRAFT NoC

Chakib NEHNOUH, Mohamed SENOUCI
Department of Computer Science
Faculty of Engineering, University of Oran1
Ahmed Ben Bella, Oran, Algeria

Abdelkader Chaib
Department of Computer Science
Faculty of Engineering University
of Tiaret- Algeria

***Abstract*—Many fault tolerance techniques have been proposed in Network on Chip to cope with defects during fabrication or faults during product lifetime. Fault tolerance routing algorithm provide reliable mechanisms for continue delivering their services in spite of defective nodes due to the presence of permanent and/or transient faults throughout their lifetime implementation. This paper presents a new approach in the domain of fault-tolerant NoC with two main contributions. Firstly, we consider a unified fault model that include transient faults, permanent faults and congestion considered as a fault. Secondly, we present a new architecture based on sub-nets and give an overview of the associated test and (re)routing algorithm. The main result of this paper, is a new routing algorithm called Collaborative Routing Algorithm for Fault Tolerance in Network on Chip (CRAFT-NoC). We compare our approach with ACO-FAR that considers as well congestion and permanent faults. Our simulation results show significant improvements in terms of both latency and reliability.**

***Keywords*—Network on Chip; Fault Tolerance; Congestion; Reliability; Sub- network; Routing Algorithm**

I. Introduction

Network on Chip (NoC) has emerged as an efficient architecture to manage communication in 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 performance and power concerns. Decreasing transistor size also rendered semiconductors more sensitive to faults and leads to serious reliability concerns in the NoC. Commonly there are two types of faults that can occur in network on chip: permanent faults (or hard faults), and temporary faults (or soft faults). Temporary faults are classified in transient and intermittent cases.

Permanent faults are due to two major effects: The increasing complexity of chip manufacturing gives rise to higher rates of post manufacturing defects caused by inaccuracies of the photolithographic and etching processes, leading to variability of material impurities, doping concentrations and size, and geometries of structures[1]. On the other hand, decreasing feature sizes cause faster transistor aging and eventually transistor wear out, caused by Hot Carrier Injection (HCI), Bias Temperature Instability (BTI), Electro-migration, and Time Dependent Dielectric Breakdown (TDDB) [1].On another side, soft errors are apt to occur at any time during the normal operation states of the system and affect randomly any part of the system. They can be forecasted and treated during runtime. The majority of failures (80 %) are caused by transient faults, whilst the rest of them originate mainly in permanent and intermittent faults [2].

Faults in different components of the NoC have different causes, however, all can result in cruel consequences: loss of packet data, misrouting, deadlocks, to incorrect functionality. Hence, the reliability of communication becomes an influential concern when designing the NoC. Which pushed the designers to elevate the problem of tolerance to faults. This issue also affects link and router of NoC that must require a specific attention, in order to maximize yield and to ensure correct operation. This emphasizes the significance of robust design solutions and has led to fault tolerance becoming a fundamental design constraint [3]. In this context, many fault tolerance techniques have been proposed at several levels (circuit/system and hardware/software) for critical applications. It is, therefore, essential to consider, the management of failures, ensure correct and continuous operation of the circuit in its environment, even when the failure rate is high.

Considering the problem above, many relevant fault-tolerant routing algorithms have been proposed, while they didn't consider, the load-balancing of network [4]. Analyzing the state of art, the objective is to design a new routing algorithm which will not only be fault tolerant, but also we recognize the network congestion state to improve the routing performance by adaptive path selection.

The authors of [5] propose an adaptative routing algorithm which measures the congestion level of the regions near the router by RCS (Regional Congestion Status), and finds a low congested path by selecting less congested links. The proposed algorithm accomplishes a good gain for reducing load-balancing and latency by applying RCS. Though, this solution is not implemented with fault tolerance mechanism.

The adaptability viewpoint, us can classify routing algorithms into two categories: deterministic or adaptive. A deterministic routing algorithm uses a fixed path for each pair source-destination node and does not consider, the current network status, resulting in increased packet latency and especially in the congested networks. On the opposite, adaptive routing algorithm estimates, the state of the network for generates multiple paths between each source-destination pair.

Other classifications are done considering where and how

[1] International Roadmap Committee. 2014. www.itrs.net

the routing decision is taken. Sometimes the characteristics of the path determined by a routing algorithm are considered relevant; thus, there are minimal and non-minimal path routing algorithms. The former is usually using the shortest one, so it generally incurs lower latency[6]. Despite, this is not always the case, for example when we have a congestion state or faulty link/router appear along the minimal path.

The idea is to couple adaptive methods designed for energy efficiency with the use of redundancy to get reliability. Adaptive methods track energy efficiency by activating NoC resources according to communications requirements, we propose a unified solution where we activate resources to solve faults including congestion.

In this paper, we present CRAFT-NoC, a new architecture for NoC . This solution offers reduced latencies and enables the use of alternative paths when necessary; The proposed work aim to jointly address congestion management and fault tolerance. The proposed solution collects the global congestion information for each subnet and adjusts path selection in the network by measuring local congestion status for each node. A shorter latency can be achieved by applying our routing algorithm. Besides, we add a fault tolerant mechanism to handle link or router failure by relying on alternative paths. Moreover, our routing algorithm is deadlock-free and finally, we verify and analyze our approach with Nirgam simulator[2]. Pure software faults are out of the scope of this research.

The rest of the paper is organized as follows. Related work is presented in section 2. The architecture of the proposed solution is presented in section 3. Implementation details of the proposed solution are given in section 4. In Section 5, CRAFT-NoC is evaluated. Finally, conclusions are provided in Section 6.

II. Related Work

Reliability can be measured and ensured through testing and fault tolerance. Testing defines the reliability of the circuit with respect to manufacturing defects. Fault tolerance ensures the reliability with respect to faults that appear during the system normal operation. Both aspects need to be considered in the NoC and in the NoC-based SoC [7]. FT approaches are usually divided into two categories: reactive and proactive techniques. The former, which can be most effective after the system is affected by the error. The latter can be used to prevent or avoid errors before they occur.

Applications communication can be critical and requires a higher degree of reliability. Many solutions have been proposed in the literature to sustain the reliability of NoCs, including component redundancy, reconfiguration, and retransmission techniques or fault-tolerant routing algorithms. But most of them focused only on one type of fault. For example, the routing algorithm proposed by Zhang and al [8] can tolerate only one faulty router. For other works [10], [17] the routing algorithm can't detect or tolerate unreachable destinations.

Redundancy is the best-known, fault tolerance technique and was the simplest method to achieve reliability. However, using this technique proposed in [9], [10], [16], [11], [12] is specially used to avoid faults in links or routers, when a component fails it is simply replaced by its copy. The disadvantage of this solution is that it is more expensive. Another drawback of redundancy is that it is sometimes necessary to sacrifice healthy routers to keep a regular area.

Others solution use retransmission [9], [13], [14], [15]. Park et al[14], propose a new technique to tolerate transient errors. They introduce retransmission of flits for detection and who are temporarily corrupted, they assert that the proposed solution has lower overhead compared to other work. Another work proposed in ARIADNE network [9], uses up*/down* routing to move around faults. After each time, when faults are detected, the new routing paths are created by transmitting a series of flag broadcasts to all routers. The disadvantage of this technique is the consumption of bandwidth which will decrease the throughput and increased the latency.

By applying the reconfiguration mechanism [16], [17], [18] new topology will be discovered and the components of the network are updated to compute the new routing path. The solution proposed by Zhang, et al[8] enforces with this mechanism. This solution requires that the defective routers (creating holes in the network) will be located accurately. Later a communication infrastructure to will be reconfigured the routers surely. The 2D DSPIN networks introduce a configuration register into the routers that allow the modification of the X-first routing by default.

For this technique, the problem is either to reconfigure the neighboring routers to create zone bypasses [17], or to stop them and restart the application. In the latter case, this can interrupt the normal operation of the system and stop the delivery of packets. Also, a good fault-tolerant routing algorithm should ensure its operation without disruption of the network. Added problem is when the reconfiguration process will be fail in a router it can disrupt the functionality of all the system or a part of it. Nevertheless, to reduce latency, a good routing algorithm will be better than retransmission and reconfiguration.

Some of them use an adaptive routing algorithm to route the packets around a faulty nodes or links [19], [13]. I. Pratomo et al [19] propose adaptive fault-tolerant routing algorithm for 2D mesh called Gradient, this algorithm is not deadlock free. Hsien-Kai Hsin et al[13] proposes a new adaptive routing algorithm called (ACO-FAR), that is biologically inspired by the behavior of ants to achieve fault-tolerance in the NoCs. Another solution proposed in Vicis [10] network, who changes its routing algorithm to circumvent faults when they are detected and turn restrictions are placed to avoid deadlocks. The disadvantage of these algorithms is that they allow to tolerating only the permanent faults.

In [15], authors present online fault-tolerant routing algorithm for 2D Mesh Networks on Chip. The proposed solution works by exploiting local information about the state of links and routers. Self-checking is used to detect faults in them. In a case of error, flit retransmission occurs from the upstream router. The messages are protected by ECC. In the presence of runtime errors, packet retransmission combined with novel message recovery mechanisms are utilized in order to provide fault tolerance under high failure rates. they have shown, that the proposed algorithm maintains high reliability of more than

[2] nirgam.ecs.soton.ac.uk [Online; accessed April- 2015].

99.38% in presence of 384 simultaneous link faults.

The disadvantage of all routing algorithms cited above [15], [17], [13], [9], [10] is the large overhead, which can generate a high energy consumption.

All cited approaches in the discussion, have benefits and drawbacks. The problem is that all these techniques have a cost in terms of performance, for instance: latency, an overhead of area, throughput, network congestion and energy consumption. Thus, it is better for the designer to find a good trade-off between these costs and reliability.

To our understanding, this is the first work that can provide all the requirements of the fault tolerance. Online detection and isolation for permanents, transients faults. Secondly, the routing algorithm ensures the delivery of packets to its destination when a path exists, as it can indicate if the destination is unreachable, it offers complete coverage. In extension, routers do not require any virtual channels and work in a fully distributed way to transmit the packets in case of failing nodes.

III. Proposed Network Architecture

Segmentation is based on the concept of maintaining connectivity to circumvent defects. A sub-network can be described as a set of interconnected links and routers, which each IP (Intellectual Property) is connected via a single link with the other sub-networks.The global architecture is depicted in Fig 1.

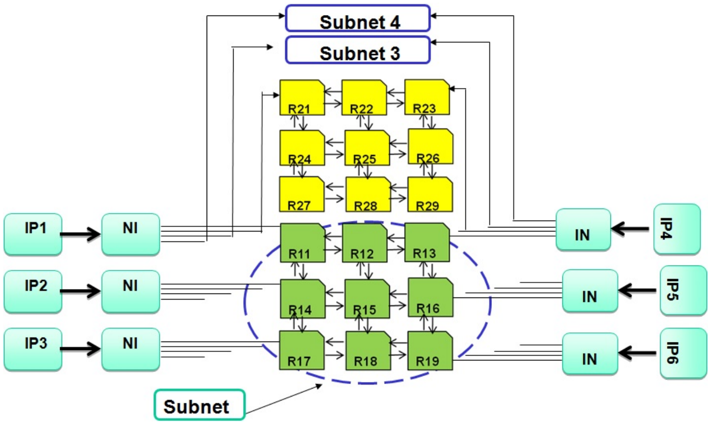


Fig. 1. The Global Architecture of NoC

A. Sub-network

There are many topologies that have been proposed for Network on Chip like Mesh, Torus, Star,..etc. In this paper, we suggest taking advantage of a topology based sub-networks that can be switch on/off according to bandwidth requirements as introduced in [20]. In this approach, we consider such an energy-proportional architecture as a global solution to deal with any type of faults including temporary, permanents faults and, congestion which prevent the system from delivering the expected quality of service (QoS). This approach allows tracking, with the same mechanisms, the best energy efficiency with or without a presence of faults. It also offers a simple solution to manage critical (no data loss) and best effort (possible data losses) communications. Fig 1 shows the CRAFT architecture:

1) the connection pattern between switches in the same sub-network,
2) the connection pattern between switches and IP cores. Every IP core is connected to four switches each one belonging to a disjoint sub-network.

Notice that switches of different sub-networks are not connected between them. Specifically, we have four SNs. Each SN is used only when necessary, otherwise only the subnet 0 is ON in the first time and all others sub-networks is in OFF state.

B. Network structure

Any 2D-Mesh network with any size can be constructed using the structure cited above. Fig 1 show an example of 2D network with 6x6 dimensions which is designed using four SNs.

C. Setting up the router according to its SN

To identify each router, we defined 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, the table below shows the codes associated with the different SNs:

TABLE I. The Codes Associated with the Different SNs

ID SN	Code
0	00
1	01
2	10
3	11

D. Router architecture

Two fundamental components are added to the basic router architecture shown in Fig.2.

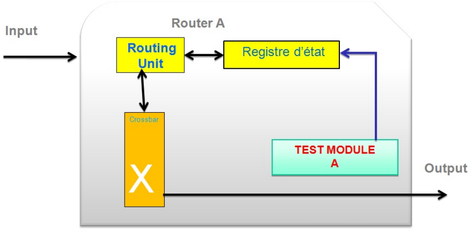


Fig. 2. Communication between Test Module and Fault Register

- Test Module :its, role is the input/output signals receiving for propagating fault information between the adjacent nodes. More details about the Test Module, fault detection mechanism is be given in Section IV. Thus, compared with the baseline router architecture, an additional multiplexer (MUX) is added and controlled by the Test Module to provide all different

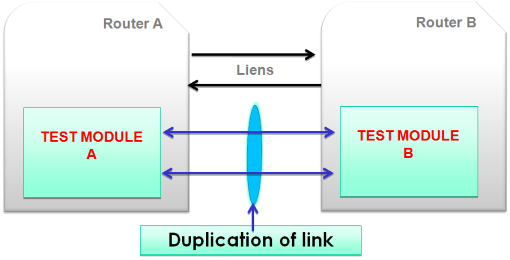
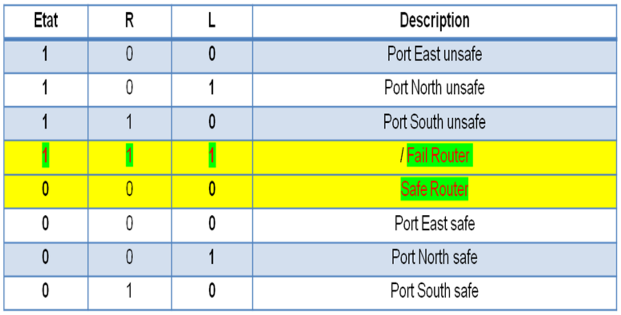


Fig. 3. Communication between two neighboring routers

signals and transmit this information to a neighbor router, therefore the two modules communicate with them. As indicated in Fig .3 we have to duplicate the links if a link is broken, we can use the second one, but not both at the same time, in order to communicate the defective components of the current router and to update the fault register to the adjacent node.

- Fault Register: Routing function unit evaluates the candidate channels in function the received or stored Fault Register information, and chooses a proper output channel to sends them. Notification mechanism is implemented with local signal connections with neighboring routers. The implementation of the FR value is twofold. First, it can be stored in each router. The area overhead a 3-bits table per router (four directions) see Fig. 4. On the other hand, in this paper, as shown in Fig. 2, we set the FR as a wiring signal for each router separated from the link connections.



Etat	R	L	Description
1	0	0	Port East unsafe
1	0	1	Port North unsafe
1	1	0	Port South unsafe
1	1	1	/ Fail Router
0	0	0	Safe Router
0	0	0	Port East safe
0	0	1	Port North safe
0	1	0	Port South safe

Fig. 4. Codification of different states of links and routers

The routing algorithm is based on this architecture. The differences are on: the Fault Register information and Test Module.

E. Subnet state and Retransmission policy based on criticity

The table .3 below shows the different types of state that each subnetwork can have with the coding of each state:

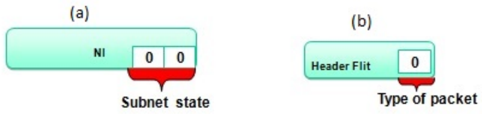


Fig. 5. (a) Subnet state,(b) Type of Packet (TP)

TABLE II. Coding of the Different States for Each Subnet

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

Indeed, we distinguish two main types of packets: critical packets and non-critical packets. In Fig 5, T.P: defined on a 1 bit, it indicates the type of packet, indeed this bit is 0 if it is a non-critical packet otherwise it is 1. This information is added to Header Fit. In this paper, we focus on tolerating permanent and transient faults in NoCs, based on detection and retransmission, for the transferred information not all data is equally critical. Specific codes can be designed to reduce the overhead by protecting the most critical data (see IV-D).

IV. Routing Algorithm

The sub-network routing algorithm (SR) is a routing algorithm, that computes alternative paths based on local or regional information for the transmission of packets in a network. It divides the entire network into subnets. Each subnet contains the same number of the router. At the same time, it is restricted to provide a greater degree of freedom and tolerance for wrongdoing.

For Fault Tolerance (FT) aim, it is very attractive to use others sub-networks. If a fault is detected at a given link (routers, wires, buffers) for one subnet, there is an alternative path that is capable of preserving the communication between PEs.

This section presents the need for a step-by-step approach to obtain an FT-NoC, and summarizes each step for this approach. For example, we assume that a fault occurs in the path between two cores. For this reason, to avoid system failures caused by hardware faults, the system must detect, isolate and avoid the defective nodes. So, the system must support adaptive routing for delivering packets using an alternative path.

This approach adopts a 2D-mesh topology, with input buffering, credit-based flow control, and wormhole packet switching. The routing algorithm between PEs combines the two distributed routing algorithm North last and South last[22]. The present work assumes only permanent and transient faults in link/router, others components are out of the scope of the current work.

A. Congestion detection

The performance of a NoC is related to the management of congestion when the traffic increases or exceeds a certain level, the latency increases and thus, the throughput decreases. The reason is that when traffic increases, several packets competition for access to the same resources.

The management of congestion is, therefore, unavoidable and its implementation is multi-constrained: Firstly, its implementation time, with a low surface cost and less consumption.

For adaptive routing, several paths can be considered when transmitting the packet and the selected path is the least congested, Thus the traffic loads are congested around the faulty nodes. We integrated congestion state to evaluate and relieve the traffic, local and regional congestion status to select the better path. We consider that the congestion condition is more severe when the number of faulty nodes increases. The congestion process is through a single additional bit in the buffer at each node, which is the minimum requirement for detection. This bit is used only when congestion is presumed and can be accessed from all neighboring nodes in the same subnet.

Inspired by Catnap [20], we propose a local congestion metric called the maximum buffer occupancy (BFM). Occupancy of a routers input buffer is the number of flits in that buffer, and it is proposed in this policy after evaluating several other policies that looked promising, In Catnap, the NI at a node keeps track of the buffer occupancy of each routers input buffer. They chose this metric for two reasons: it is independent of the network traffic pattern. Also, it incurs lower design complexity than the other alternatives.

The LCS shall be designed to sense this local congestion condition for early detouring. To achieve this, one bit is added for each buffer that collects and propagates the information of congestion in each router.

1) Local Congestion Status (LCS)[20]: If the BFM of a router is greater than a threshold, then that routers subnet is considered to be congested, and a local congestion status (LCS) bit is set 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,
2) Regional Congestion Status (RCS)[20]: 1-bit OR network that collects the congestion status of all the routers in a region of a subnet. This bit value, which we refer to as the regional congestion status (RCS), can be read by all the routers in the same subnet .The OR-network is architected as an H-Tree network. The NI of a node sets its RCS if its local congestion status (LCS) is true which is determined based on the BFM of its local router (that is, anyone of that subnets routers in its region is congested). A nodes NI detects 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.

B. Fault Detection

Fast detection becomes a necessity, and the use of on-line tests becomes essential, where network components (eg network link) become unavailable and this must be done in a periodic and frequent manner during the operation of the system. The intention is to use CRC to detect faults and to be able to pinpoint the location of each defect and finally use this information to update the fault register. One important feature of the Test Module is the fact that the isolation is decoupled with the fault detection. So, the main function of this Module is just to write in the fault register when it detects a defective router or link. According to Figure 6 the Test Module is to cope with detecting faults in three different locations:

- Fault in the link itself (wires) and input buffer;
- Fault in the crossbar of switch;
- Fault in the header flit.

Then, to prevent the spread of faults, isolation ensures that the defective area does not disturb the neighborhood, and all incoming packet will be immediately deleted. This Module requires minimal extra hardware. Moreover, it is possible to shut off one router or disable link and can't reduce gracefully the network performance when the number of faults increases. Therefore, if some routers fail, a new path may be used to route the packets from source to destination in the same subnet or by another.

At first, the fault detection mechanism uses Test Module the particular circuit to detect, locate, and isolate the faulty in routers or links, Therefore, only adjacent routers can notice the fault in the same subnet . Figure 6 presents the approach inspired by [21], which uses CRC decoders to detect faults. The router can receive CRC decoders in the following locations:

1) Before the input buffer (CRC 1), with the objective to detect faults in the link.
2) After the buffer (CRC 2), with the objective of detecting faults in the buffer. The channel can be healthy, but a fault can change the state of a given bit stored in the buffer.
3) For detecting internal errors of the router, we use CRC 3 with the objective to detect a fault in the crossbar. Moreover, in this case, the entire router should be disabled because the integrity of the packets cannot be guaranteed.
4) The CRC4 was added to detect faults at Header flit that may occur during the transit of one package. These faults can potentially lead to network deadlocks due to poor routing. When this kind of error is detected, it is considered as a critical failure.

When the fault is detected by the fault detection mechanism, different signals f_i value is sent to the router adjacent to the faulty node (see Fig 3). F_{out} is the signal propagating from local router to neighboring routers, in order to update FR (Fault Register), We can observe from Fig. 2 and 3 that the adjacent router of the faulty node can receiving three signals F_{out} = 3. To make routing more efficient, the upstream router react correspondingly depending on the received F value to reroute the packets. Hence, this can bring the traffic load away

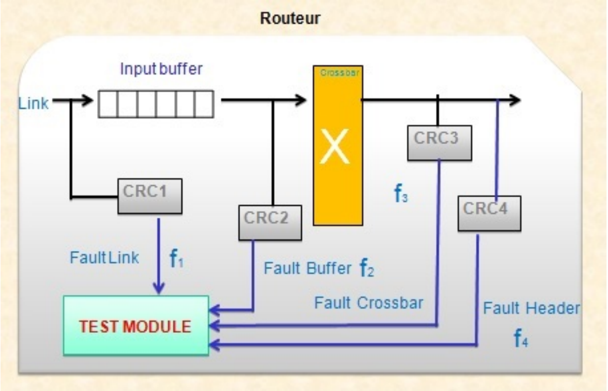


Fig. 6. Internal architecture of the router

from the faulty node and reduce the congestion of nearby routers.

Defects in an integrated circuit can be classified into three categories according to their behavior: permanent, transient and intermittent.

- Permanent Fault Detection: Permanent errors are due, for example, to disturbances in the manufacturing process or to phenomena of aging of the circuit. These errors cannot be eliminated by a simple reset of the circuit. Routers adjacent to the router with a permanent fault are notified of Its state and this, to prevent any traffic to this defective router (for example, they can disable the output ports leading to this router). Same case for defective links.
- Transient Fault Detection: Transient failures are due to temporary external environmental events. These errors typically occur during a very short time and are not destructive. So, after k attempts, the test module can consider the (temporary) dynamic faults as permanent faults. So we want to tolerate several faults on-line and without having to reset the circuit, for this, the adaptive routing is, therefore, the most suitable, in the presence of defective or congested links/routers.

C. Routing Algorithm

In this approach, the isolation is strictly coupled with the FT-NoC routing, that mean when a fault is detected in the link or router, for example, the whole router is disabled (link).The objective of a novel fault tolerance routing is to find a new path for every source-target pairs in a faulty network. So, the routing algorithm may require a turn for avoid deadlock and circumvent faulty nodes in the presence of faults. Thus, a fully adaptive routing algorithm is required. Any 2D-mesh can be divided into four disjoint sub-networks, each one implementing an adaptive routing algorithm, for example, North Last and South Last[22].

Many scenarios are adopted and when faults are detected the distributed routing is applied using North Last and South Last to reach the destination. However, area and power consumption is still an issue in the resource-limited NoC, the hardware cost of routers is a critical issue, so VC can increase significantly the area cost and power consumption, for this reason, in this paper new routing algorithm are proposed, without using virtual channel to achieve fault-tolerance, and turn model to guaranteed deadlock free.

At the system startup, the network is supposed faulty-free, and packets are sent from the source PE to the destination PE. The path searching mechanism searches the path to adjacent nodes except a faulty router or link in the same subnet or others subnets to provide higher path diversity.

Base on 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 reachable or not reachable, we also illustrate these cases by using Fig 8.

- Congestion case; in the first scenario if the source PE identifies it is not able to transfers packets to target PE. The routing algorithm provides a new path, by switch-on the higher level. In this case, set the current subnet state congested and the new packets can be injected in new subnet.
- Fault case and destination reachable; In this case, the path is faulty, the faulty router adjacent to the current router (received packet). The routing algorithm provides the next hop by applying the appropriate turn North last or South last depending to link and router state saved in Fault Register (FR).
- Fault case and destination not reachable; in this case path prohibited. The routing algorithm provides a new path, by switch-on the higher level and set the current subnet state to under broken, in this case, all new packets can't be injected in the old subnet.

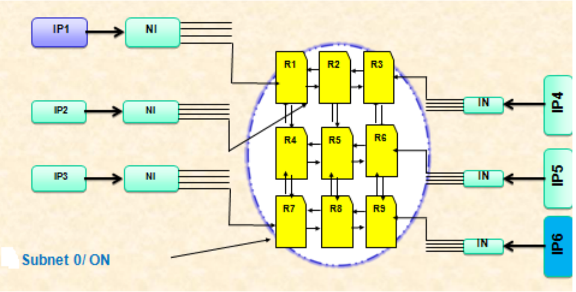


Fig. 7. Congestion case : Subnet 0 is congested

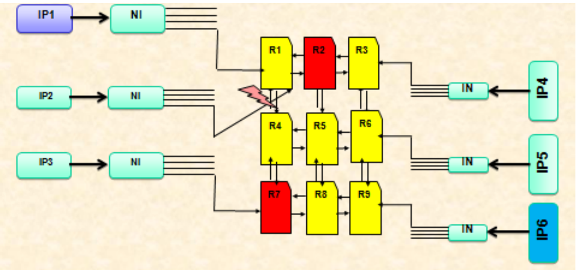


Fig. 8. Routing algorithm in case II and III

D. Retransmission of packets

In the case of a non-reachable destination, the retransmission of the flits is initiated by the upstream router. The retransmission mechanism is made according to the type of packet, if the packet is critical the retransmission is made if not lost.Thus, we keep a copy of the header file for critical packets at the router before transmitting it. So, this mechanism must be implemented by the source node. The algorithm of routing suggested sends only the critical packets. In our case, each PE its network interface (NI) and the links linking the router to the NI are considered healthy.

However, this mechanism aims to tolerate dynamic faults (temporary) during the transit of the packet that modifies the validity of the path, For example, a router or link becomes suddenly defective while all the flits are not passed, thus creating sub-packets which cannot all arrive at their destination (Fig 10). A notification message is sent to the source node to transmit all the packets again and to drain the flits which have

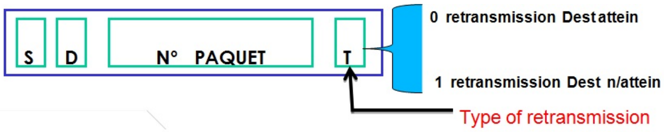


Fig. 9. The notification message format

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 below the notification message in both cases.

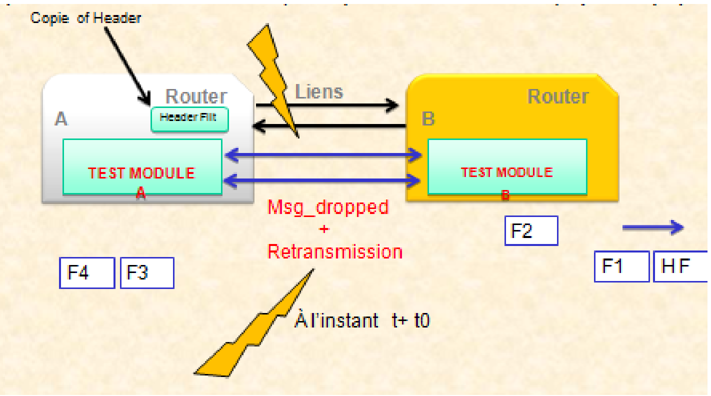


Fig. 10. The retransmission process in temporary faults case

The notification message format is presented by the Figure 9.This message differs depending on the retransmission type. Indeed, there are two types of retransmission (0: attainable,1: unattainable). T: defined on 1 bit. S, D: address of source and destination, finally n packet: denotes the sequence number of the packet. This field is defined on 4 bits.

E. Deadlock avoidance

This algorithm combines two adaptive routing algorithms North-Last and South-Last that use restrictions to avoid deadlocks [22]. The NL turn model deadlock-free routing is achieved by prohibiting two turns, dashed arrows indicate prohibited turns (Fig 11(a)). In this routing algorithm, the flit is routed in the E, W, or S directions before turning in the N direction, after can't make further turns. For the south last (SL) turn model it is similar to NL, a flit is routed in the E, W, or N directions before turning in the S direction after can't make further turns. As depicted in fig 11(b). This to diversify the paths if a packet can't reach its destination the second algorithm gives another possibility.

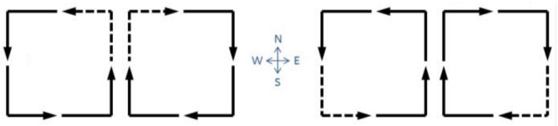
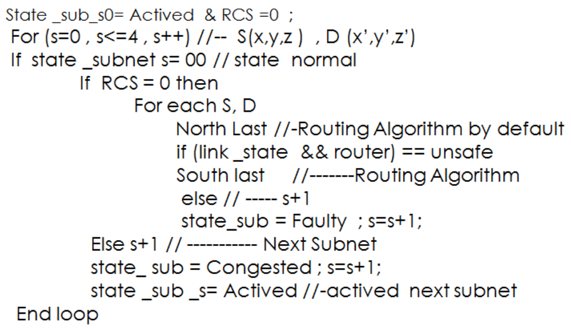


Fig. 11. Turns allowed in the (a) North-last, (b) South-last algorithms

The objective of CRAFT routing algorithm (Fig. 12) is to route S to its destination D with or without the presence of faults or congestion.



```
State _sub_s0= Actived  & RCS =0 ;
 For (s=0 , s<=4 , s++) //-- S(x,y,z )  , D (x',y',z')
 If  state _subnet s= 00 // state  normal
            If  RCS = 0 then
                     For each S, D
                             North Last //-Routing Algorithm by default
                             if (link _state  && router) == unsafe
                             South last    //-------Routing Algorithm
                              else // ----- s+1
                              state_sub = Faulty  ; s=s+1;
              Else s+1 // ----------- Next Subnet
              state_ sub = Congested ; s=s+1;
              state _sub _s= Actived //-actived  next subnet
 End loop
```

Fig. 12. Pseudocode: Implementation of routing algorithm CRAFT

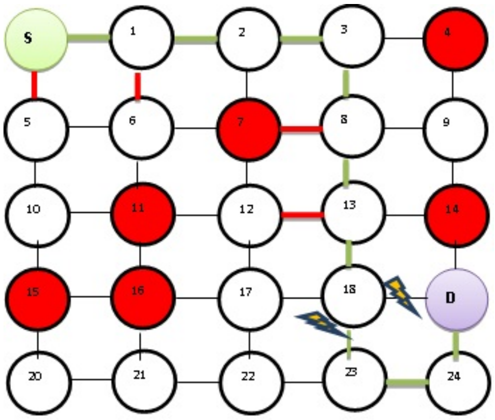


Fig. 13. Path selection scenarios in case I and II for NL turn

To understand this algorithm let us take the following example (Fig 13): Let P be a packet that is sent from a source node S to the destination node D. P arrives in the current node x.

Fig. 13 explains an example of a faulty network (red color). A packet traversing an intermediate router must choose one output directions between (S, W, E). The NL routing algorithm would take the path S-1-2-3-8-13-18-D, suppose that there are two temporary errors at the links linking the nodes (D, 18) and (18,23) at the same time. We observe that the packets can't reach the destination , so we need to send them by another subnet. In this situation, notification message is sent to source for retransmission and another message for remove all packets which are already transmitted.

V. Performance Evaluation on Faulty Network

A. Environment of simulation

Proposed routing algorithm reduces the latency and the number of packets lost for different kinds of scenarios and can be considered as a potential candidate for real application, first, we fixed the fault tolerance for our routing algorithm, so we had to adapt the configuration file for the possibility of injecting faults at the routers or links (see table 3).

The table above summarizes the possible configurations of the simulator. To measure and quantify the performance of a network on a chip, we need metrics. One of the most important criteria is latency. Secondly, reliability, and more specifically fault tolerance, Packet success rate. This rate corresponds to

TABLE III. SIMULATION PARAMETERS

Parameter	Value
Topology	10x10 , 16x16
Buffer size	10 Flits
Traffic	Uniform,Transpose
Failure injection rate (%)	0,10,20,30,40
Packet size	4 Flits
Warm-up	5000
Congestion metric	BFM
Simulated packets	50000

the number of packets arriving at their destination in relation to the total number of packets injected, this for a given type of traffic and for a certain rate of failure.

B. Performance evaluation

1) Latency: We evaluated the average latency under the different types of traffic: To compare the performances, we considered the case of a network of size (6x6), and a network of size (8x8). The average latency of each network was measured by considering uniform traffic. The calculation of the latencies was based on simulations carried out using the simulator Nirgam. The results are given by the Fig.14. To show the performance improvement of CRAFT, we evaluate our routing algorithm in uniform traffic patterns. In the experiment, the threshold T is set as 90 % of the buffer size.We also based on congestion aware.

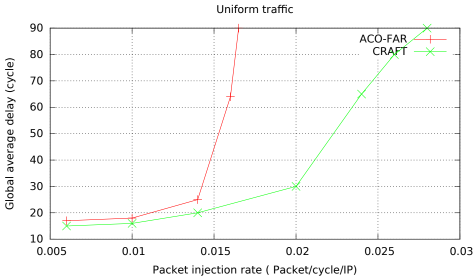


Fig. 14. Performance of CRAFT routing algorithms under 4 faulty routers and links, with uniform traffic (8x8).

The experimental results are shown in Fig.14. The performance of the routing algorithms is evaluated in terms of average packet delay. The results obtained in Fig.14 show considerable performance regarding latency, and these results are better compared to [13]. The percentage of failure rate is fixed to 40% (link and router).

2) Reliability: We evaluate the fault-tolerance ability with the delivered packets ratio . This index indicates the success rate that represents the percentage of packages which arrive at their destination in relation to the injected packets.

$$\text{Success ratio} = \frac{\text{Total. arrived packets}}{\text{Total injected packets}} \times 100$$

TABLE IV. COMPARAISON OF SUCCESS RATIO % WITH ACO FAR ROUTING ALGORITHM

No of Fault	Gradient	ACO FAR	CRAFT
2	2,8%	0,07%	00
4	3,2%	0,5%	00

This phase consists of conducting fault injections campaigns and comparing our approach with the reference [13] algorithms for uniform traffic.

Is shown in Figure 14 and Table IV, the strength of the present routing algorithm CRAFT is confirmed throughout the experiments, that this achieves shorter average packet latency compared to ACO FAR routing algorithm in presence of faulty routers and links.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new adaptive algorithm fault aware and congestion-aware for NoCs. To achieve the proposed solution, the NoC architecture is partitioned into subnets. Each one, avoids congestion state by local and regional information, to identify the best path to route packets. In order to react dynamically to the different faults in the NoC, the procedure is invoked periodically to detect and isolate the faulty components. Results based on Nirgam simulator, demonstrate that the proposed adaptive routing algorithm improves significantly the network latency and reliability, compared to ACO FAR adaptive routing algorithm. We have also proposed a new architecture for preventing the loss of packets in a critical application. Our next works include the hardware overhead, consumption energy and computational time to detect permanents and transients faults.

REFERENCES

[1] Radetzki, M.; Feng, C.; Zhao, X.; Jantsch, A. ”Methods for Fault Tolerance in Networks on Chip”. ACM Computing Surveys, vol. 46-1, 38p, October 2013,.

[2] Teijo Lehtonen, Pasi Liljeberg, and Juha Plosila. ”Online reconfigurable self-timed links for fault tolerant NoC” VLSI Design, 2007.

[3] Sebastian Werner, Javier Navaridas, and Mikel Lujan. 2016. ”A survey on design approaches to circumvent permanent faults in networks-on-chip.” ACM Comput. Surv. 48- 4, Article 59 , 36 pages, March 2016.

[4] S. Jovanovic, C. Tanougast, S. Weber, and C. Bobda, ”A new deadlock-free fault-tolerant routing algorithm for NoC interconnections”, in Proc. Int. Conf. Field Program. Logic Appl., p.326-331, Aug.Sep. 2009.

[5] P. Gratz, B. Grot and S. W. Keckler, ”Regional Congestion Awareness for Load Balance in Networks on Chip,” In International Symposium on High Performance Computer Architectures (HPCA), p. 203-214, 2008.

[6] Ebrahimi, M.; Daneshtalab, M.; Plosila, J.; Tenhunen, H., ”Minimal-path fault-tolerant approach using connection-retaining structure in Networks-on-Chip”. In: NOCS, 4p, 2013.

[7] Cota,E.; Amory, A. M.; Lubaszewski, M. S. ”Reliability, Availability and Serviceability of Networks-on-Chip”. Springer, 209p, 2012.

[8] Z. Zhang, A. Greiner, and S. Taktak, ”A reconfigurable routing algorithm for a fault-tolerant 2-D-mesh network-on-chip” in Proc. Design Autom.Conf. (DAC), p.441-446, 2008.

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

[10] D.Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, ”A highly Resilient routing algorithm for fault-tolerant NoCs” in Proceedings of the Conference on Design, Automation and Test in Europe, p.21-26, 2009.

[11] W. Tsai, D. Zheng, S. Chen, Y. Hu, ”A Fault-Tolerant NoC Scheme using bidirectional channel,” 48th ACM/EDAC/IEEE Design Automation Conference (DAC), p.918-923, June 2011.

[12] M. Ebrahimi, M. Daneshtalab, J. Plosila, H. Tenhunen, ”MAFA: Adaptive Fault-Tolerant Routing Algorithm for Networks-on-Chip” DSD , p. 201-207, 2012.

[13] Kai Hsin, En-Jui Chang, Chia-An Lin, and An-Yeu (Andy) Wu,"Ant Colony Optimization-Based Fault-Aware Routing in Mesh-Based Network-on-Chip Systems" IEEE transactions on computer aided design of integrated circuits and systems , VOL. 33, NO. 11, p.1693-1704, November 2014

[14] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das, "Exploring fault- tolerant network-on-chip architectures" in IEEE Dependable Systems and Networks, p. 93-104, 2006.

[15] M. Dimopoulos , Y. Gang , L. Anghel , M. Benabdenbi , N. Zergainoh , M. Nicolaidis,"Fault-tolerant adaptive routing under an unconstrained set of node and link failures for manycore systems-on-chip", Microprocessors Microsystems, v.38 n.6, p.620-635, August 2014.

[16] Z. Zhang, A. Greiner and M. Benabdenbi. "Fully Distributed Initialization Procedure for a 2D-Mesh NoC, Including Off Line BIST and Partial Deactivation of Faulty Components" In Proceedings of the 16th IEEE International On-Line Testing Symposium (IOLTS10 Greece) p.194-196, 2010.

[17] Wachter, E.W.; Erichsen, A.; Amory, A.M.; Moraes, F.G. "Topology-Agnostic Fault- Tolerant NoC Routing Method". In: DATE, 6p, 2013.

[18] F. Chaix, D. Avresky, N. Zergainoh, and M. Nicolaidis, "Fault-Tolerant Deadlock-Free Adaptive Routing for Any Set of Link and Node Failures in Multi-cores Systems", Proceedings of the Ninth IEEE International Symposium on Network Computing and Applications (NCA'10), Cambridge, Massachusetts, USA, p.52-59, July 2010.

[19] I. Pratomo and S. Pillement. "Gradient - An Adaptive Fault-tolerant Routing Algorithm for 2D Mesh Network-on-Chips", In Design Architectures for Signal Image Processing (DASIP), International Conference, October 2012.

[20] Reetuparna Das, Satish Narayanasamy, Sudhir K. Satpathy, Ronald Dreslinski, "Catnap: Energy Proportional Multiple Network-on-Chip", In proceedings of the 40th International Symposium on Computer Architecture, Tel Aviv, Israel ISCA, 2013.

[21] Fochi, V.; Wachter, E.; Erichsen, A.; Amory, A.; Moraes, F. "An Integrated Method for Implementing Online Fault Detection in NoC-Based MPSoCs", In: ISCAS, p.1562-1565, 2015.

[22] C. Glass, L. Ni, "The turn model for adaptive routing", in Proceedings of the 19th annual international symposium on Computer architecture (ISCA '92), New York, NY, USA, p.278-287, 1992.