

End to End QoS Metrics Modeling Based on Multi-application Environment in Network on Chip

Abdelkader Saadaoui^{1,*}, Salem Nasri^{2,3}

¹Computer and Embedded System Lab (CES), ENIS, Sfax, Tunisia

²Computer Science Department, ISET Rades, Tunisia

³Computer Engineering Department, College of Computer, Qassim University, Kingdom Saudi Arabia

*Corresponding author: saadaouiabdelkader@hotmail.com

Abstract To quantitatively measure quality of service (QoS) in Network on Chip (NoC), several related aspects of the network service are often considered, such as end to end delay (EED), Throughput (Thp), Packet loss rate (PLR), etc. However, until now, no standard method of performance measurement and fewer techniques have been used to provide its definition. In fact, few papers have developed different methods to modelize QoS in NoC and provided an efficient and flexible way to monitor QoS. The originality of our approach is based on a proposition of a QoS–intellectual property module in NoC architecture to improve network performances. We implement an approach of QoS metrics modeling for NoC, using Analytic Hierarchy Process (AHP) on multi-parameter and multi-application for 4×4 mesh NoC environment. The results have shown that our QoS modeling approach is proven successful in providing a quantifiable representation. Therefore, QoS arbiter module interacts with other routers, the link utilization is balanced and network performance improved.

Keywords: NoC, QoS, modeling, dynamic routing, QoS parameters, AHP

Cite This Article: Abdelkader Saadaoui, and Salem Nasri, “End to End QoS Metrics Modeling Based on Multi-application Environment in Network on Chip.” *Journal of Computer Networks*, vol. 4, no. 1 (2017): 1-10. doi: 10.12691/jcn-4-1-1.

1. Introduction

The NoC paradigm seems to be very attractive solution for the problem of the existing and future chip interconnect [1-6].

Although NoC's researchers can borrow concepts from the computer networking domain into a chip, it is impractical to blindly reuse features of "classical" computer networks [7].

A chip employing a NoC is composed of intellectual property (IP) cores such as routers and network interfaces, connected among themselves by communicating channels [5].

Many of the applications require high throughput and performance through a regular interconnection network, so differentiated services are provided through a class of services (CoS) based QoS architecture.

QoS involves guaranteeing service levels (SLs) to traffic flows, it specifies a guaranteed parameter level. In other words, QoS measures the performance degree in a data transfer system. Also, QoS is defined as service quantification that is provided by the network to the demanding core and is estimated through its parameters as well as in grid environment [7,8,9,10]. However, several attempts were made to find a quantifiable scale for QoS measurement.

Currently we are working on the QoS metric problem for NoC-based system. For this purpose, we propose an

approach of end-to-end QoS metrics modeling based on multi-application environment in NoC. We propose an extended approach of QoS metrics modeling and analysis based on dynamic routing for multi-application environment with multi-parameters.

Therefore, researchers are looking for a projection of QoS on quantifiable space, since it is qualitative, subjective and not measurable.

After choosing our QoS evaluation parameters, i.e. after knowing the different elements that go into building the total estimating relationships, we have to determine how to quantify this information, and how to aggregate all our measurements into a meaningful metric.

AHP is helpful in capturing subjective evaluation measurements in a quantitative manner and provides a useful mechanism for checking the consistency of the evaluation measurements [11].

The paper is organized as follows: Section 2 discusses the previous work, Section 3 gives an overview of the dynamic routing techniques in NoC. QoS metrics modeling requirements are presented in Section 4. Section 5 presents the experimental results, and Section 6 concludes the paper.

2. Related Works

In this section, we briefly survey various QoS in NoC tentative definitions. Then, a study of QoS in NoC metrics is given. Finally, the goal behind our research work is presented.

2.1. QoS in NoC Tentative Definitions

QoS in NoC has received broad attention, so several research groups have focused on parameters (bandwidth, latency, packet loss...) to ensure the QoS performance in the NoC.

Reference [3], in the quality of service NoC (QNoC) project, identified four classes of service into SoC inter-module communication. The four classes are presented in highest to lowest priority on the NoC architecture as: signaling, real time, RD/WR and block transfer, successively for inter-module control signals, delay-constrained bit stream, short data access and data blocks. QNoC design reinforces traffic flow by increasing network performance through low latency and high throughput.

In parallel, some researches that studied QoS metrics for NoC inspired from macro-network, referred QoS as the capability of a network to provide better service.

Reference [12] proposed a QoS integrated in complete asynchronous NoC architecture. This architecture targets globally asynchronous locally synchronous SoC. This provides low latency service using virtual channels (VCs). However, in [13] and [14], authors combined Best Effort and guaranteed throughput services to ensure the QoS in NoC. Nevertheless, more software development is needed to further expose QoS features, such as proposed by [15], they considered an integrated hardware–software approach for delivering QoS at the application level for NoC-based platforms. On the other hand, due to the necessity to define QoS in this promising interconnect paradigm, reference [7] presented QoS as service quantification to the demanding core offered by NoC. Reference [16] defined the QoS as a manager of distance between a service provider and service requester and have normalized QoS parameters. They proposed an algorithm to normalize parameter values.

2.2. QoS in NoC Metrics

There is a need to quantitatively evaluate the QoS in NoC for different application and multi-parameter in order to make an informed decision about the state of the NoC for users and service providers.

References [17], [18] and [19] addressed the problem of metrics for end-to-end QoS management on real-time applications by presenting a virtual communication support. Their research was focused on the study of QoS through the switch buffering requirements. Reference [17] were interested in NoC switch scheduling and its impact on QoS metrics. Recently, [20] proposed a novel agent-assisted QoS-based routing algorithm for wireless sensor network applications. The proposed algorithm shows that it can ensure better QoS by improving network performance such as delay, bandwidth and packet loss parameters to increase the QoS level of network.

To the best of our knowledge, there has been very little work on the same subject. In fact, [21] and [22] proposed two approaches of QoS metrics modeling based on the QoS parameter estimation.

The approach proposed in [23] is based on the deficit weighted round robin technique for the management of the data queuing for one application.

2.3. Aim of the Paper

Without a structured method, trying to evaluate the overall QoS in NoC can be a very challenging task.

For this purpose, QoS parameters are an important metric in NoC performance evaluation.

However, evaluation parameters are difficult to formalize because of difference in their measurement units.

The QoS model approach must identify and weight evaluation parameters then collect, analyze and normalize data for each parameter and finally can estimate the cost of the model.

The main contributions in this work are as follows: we: (1) improve the work proposed in [24] by adapting the classical Analytic Hierarchy Process method to quantitatively evaluate our 4x4 mesh NoC performance, (2) propose a QoS IP module in NoC architecture to improve network performances.

3. Dynamic Routing Techniques in NoC

In this paper, we propose the use of 4x4 mesh topology as shown in Figure 1. In which, each router has a set of bi-directional ports linked to its neighbor routers and to an IP core.

In this study, we considered three different destinations connected to router 12, router 22 and router 21 where same distance of four hops is defined between sources Ap_i and destinations $Dest_i$ in NoC study case.

Routing algorithm purpose is the mechanism responsible for determining the path that a packet traverses from the source node to the destination node. [25].

Furthermore, the primary goal of dynamic routing technique is to ensure that data transferred to the network reach its destination.

Network congestion occurs when a link is full up and cannot handle any more traffic that it is carrying, so additional packets are thrown away then its quality of service will suffer and deteriorate.

In Figure 1, routers 01, 02, 12 and 22 have a routing problem (worst case) to forward a set of flits, an appropriate strategy is needed to ensure flit transfer and to prevent dropped data.

There will be severe congestion on many links and low utilization on vertical and horizontal other links. Thus, the network performance is impacted.

To show the problems of conflicts in the NoC, simulations were performed using the ISE Simulator (ISim) tool (www.xilinx.com/products/design-tools/isim.html). The testbench result of a simulation scenario of two applications related to the routers : R_12 and R_6 that send five packets [ap1(b0,b1,b2,b3,b4) and ap2(a0,a1,a2,a3,a4)] to a shared destination connected to the shared router R_3 is shown in the following Figure 2.

This result shows a loss of packet b4 of ap₁ on reception (Packet loss = 1 packet i.e $PLR = \frac{1 \text{ packet}}{5 \text{ packets}}$) and

this is due to a problem of conflict between the flows of the two applications, the saturation of the tail of the receiver router and of a lack of arbiter mechanism to manage the communication within shared resource inside the network on chip.

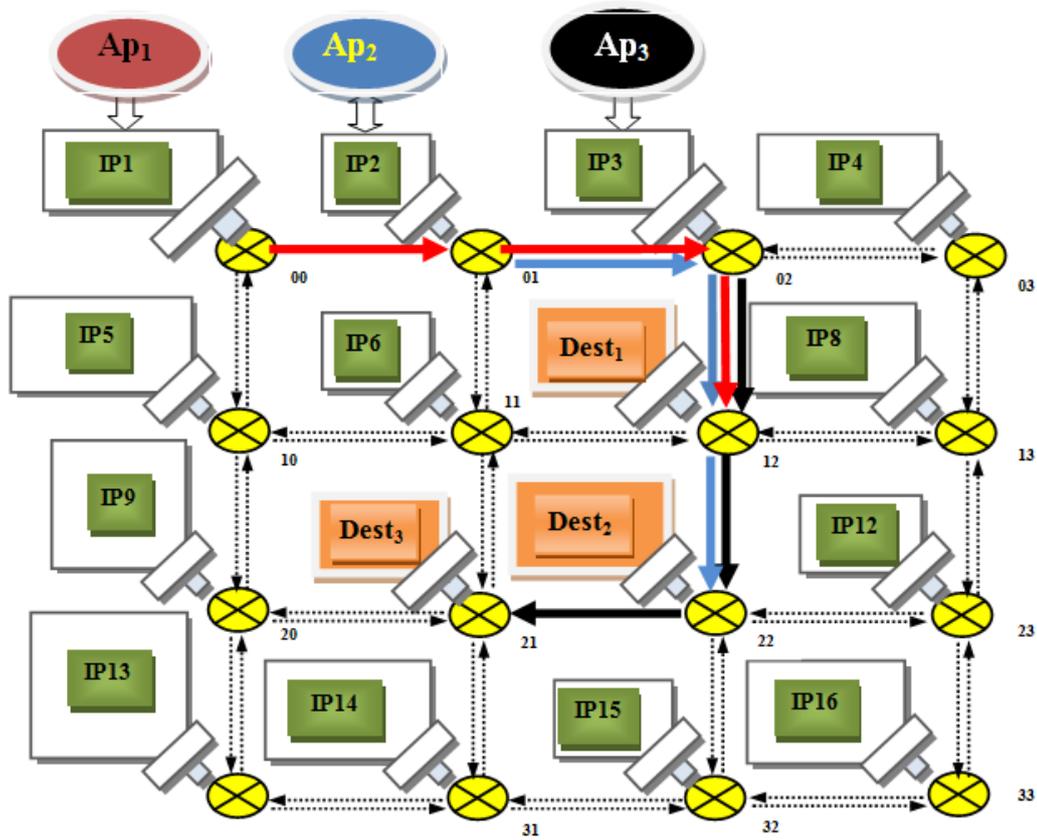


Figure 1. Routing problem in 4x4 Mesh NoC structure

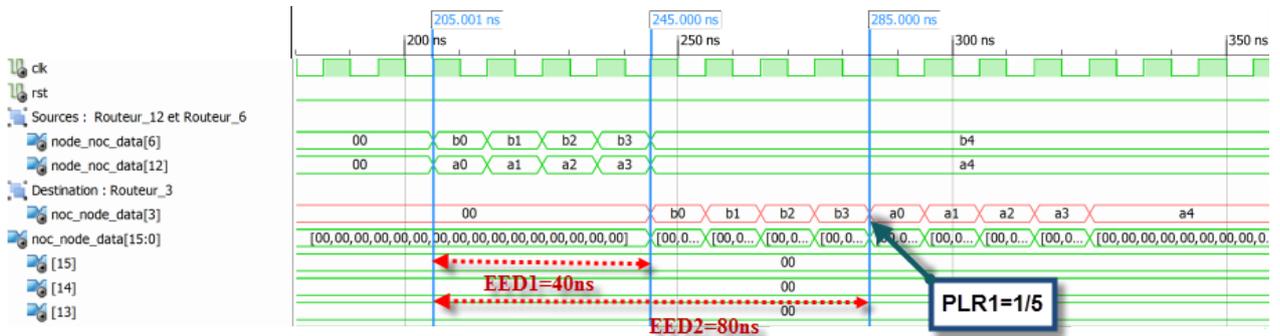


Figure 2. conflict problem in the NoCs

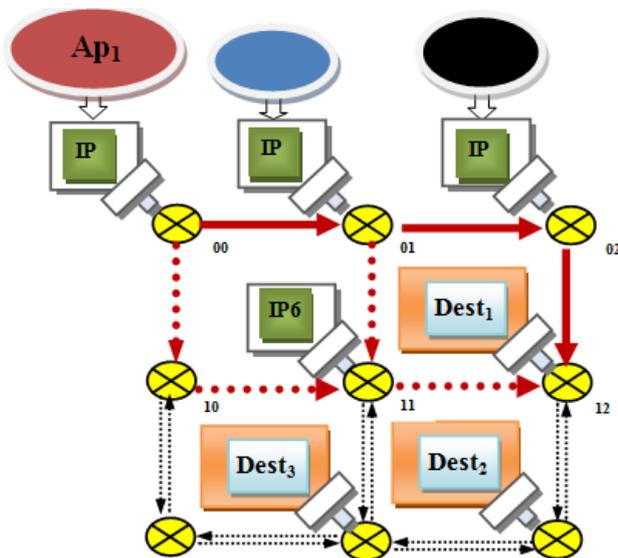


Figure 3. Example of route selection

Moreover, EED1 of ap_1 has four clock cycles, however EED2 of ap_2 has eight clock cycles as network on chip performance. Indeed, in this simulation, the propagation between two successive routers (also called "hop" in the literature) consumes a clock cycle.

Consequently, Figure 3. for example, if there is a packet being transferred from Ap_1 to $Dest_1$, there are many alternative paths from router 00 to router 12: (00→01→02→12; 00→01→11→12; 00→10→11→12).

Our methodology allows packets to take other path depending on the network state, if selected path is congested. This approach helps balance link load and relieve congestion.

3.1. QoS integration in NoC Environment

The QoS IP in NoC architecture (Figure 4) assists to improve network performance by allowing the capability of reserving routes between sources-destinations and arbitrating flits to ensure end to end QoS parameters.

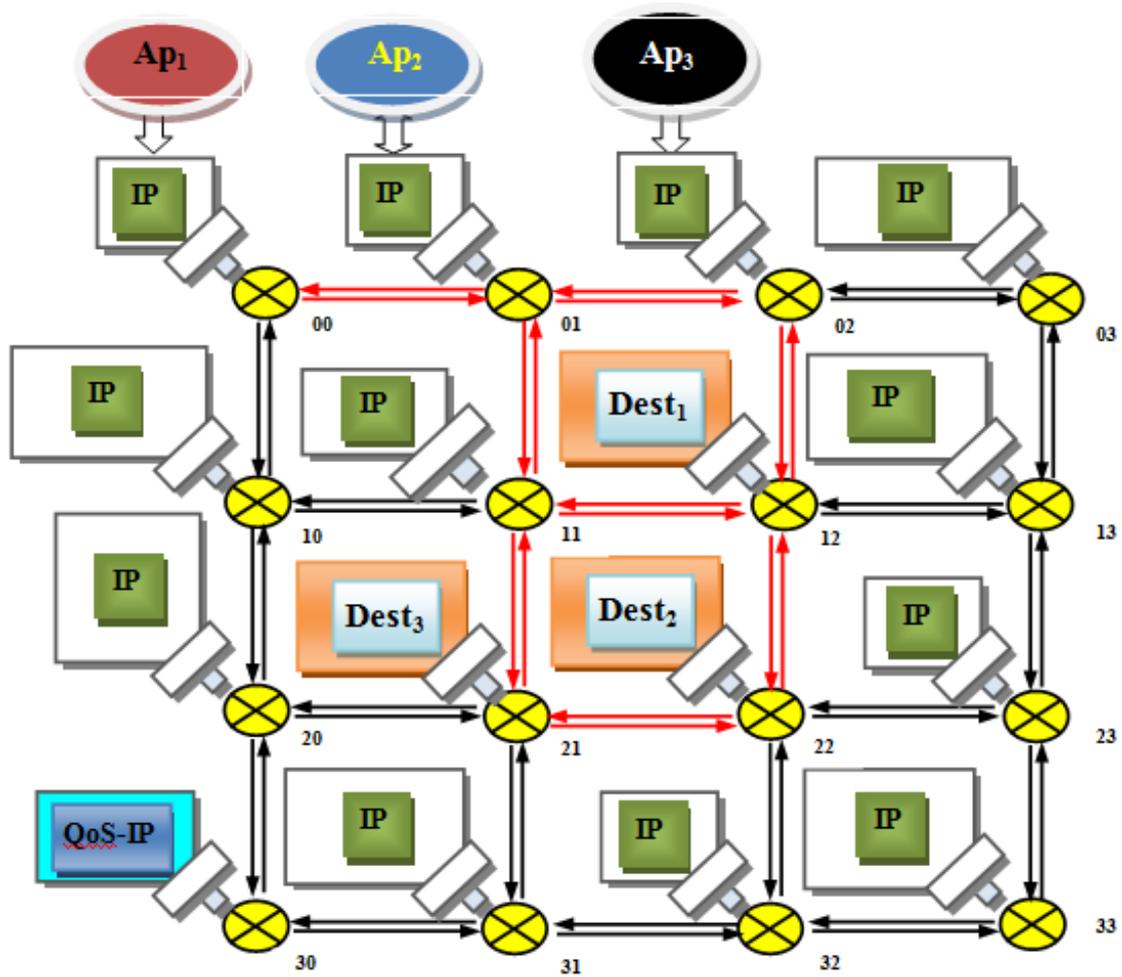


Figure 4. Example of route selection

Furthermore, the position of the QoS module inside the NoC will certainly have an influence over the amount of overhead information generated by it in the different routers of the network. Notice that we have two ideas about QoS-IP position in NoC architecture. Firstly, we can make the QoS module in a corner of the NoC, all the information exchanged with the QoS module each is in millisecond, will have to be routed by only few of network branches and the time delay for information to reach the opposite corner will be considerable, however the network is not congested by QoS-IP data.

Secondly, if the QoS-IP is located in the centre, will be possible to route information through many branches and the Euclidian distance to most distant routers will be shorter.

Unfortunately, I have not compared the influence of both on system performances which will be the subject of future work.

The fundamental problem is that all NoC resources are limited, including router processing time and link throughput. The network can easily be congested by a few ap_i flows and QoS requests, why we chose for our traffic management strategy, a different value speed between ap_i data transfer (10^{-9} s) and a data collect by QoS IP (10^{-6} s).

This speed difference used to avoid the overload of network capacity. Thus, we think that every millisecond is adequate to provide a quantifiable QoS representation.

Each application generates packets that have different priority classes in terms of delivery probability.

A flit transfer, carries a SL priority index, can be classified by higher Service Level (SL) priority and if channel is busy, QoS module can be intervening.

This study is motivated by avoiding many data transfer problems (deadlock, starvation, drop) and improving flit deliveries of high priority class, proposed scheduling policy focuses on improving flit deliveries of priority class.

3.2. Intervening Parties in QoS-IP

Figure 5 shows the intervening parties in QoS module. The role of this module can also efficiently be achieved by exchanging NoC state information.

Each application ap_i sends its data to destination $Dest_i$. Channel between routers must support, by hypothesis, too many transfer data.

To evaluate NoC performance, QoS-IP module collects data to know NoC state via router 11 and reads prioritization factors applications and parameters to provide sensible decisions based on QoS.

In fact, Figure 6 shows a priority scheme proposed in router architecture which helps to alleviate the effects of congestion for ap_i traffics.

Control signals exchanged between routers as well as QoS module, checking the NoC state and allowing dynamically the search of optimal router port or through other router using the shortest path to the destination to ensure network performance.

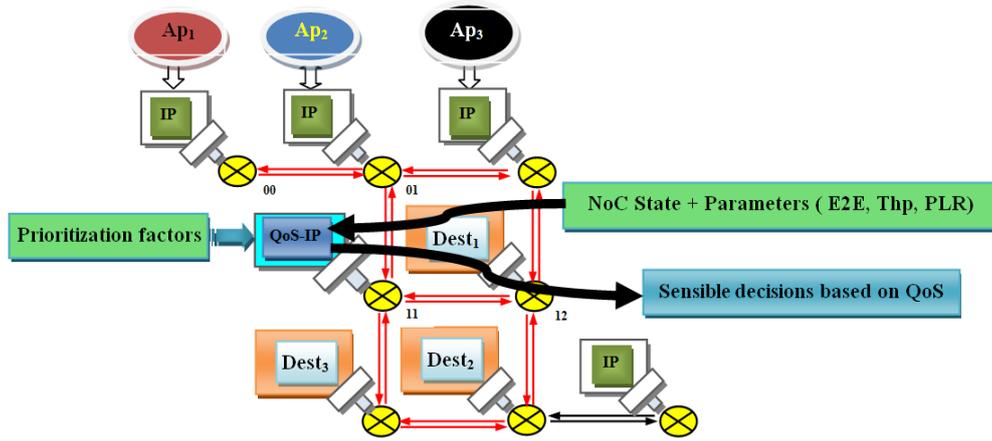


Figure 5. QoS module parameters

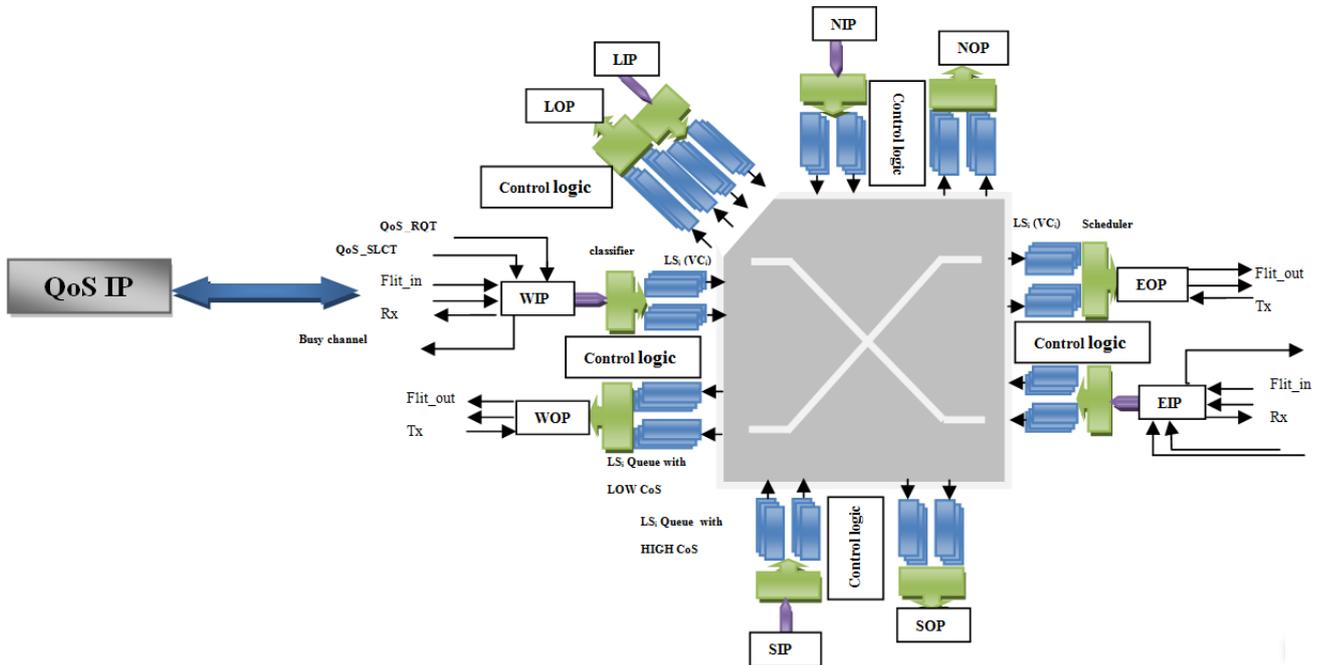


Figure 6. Priority scheme in router architecture

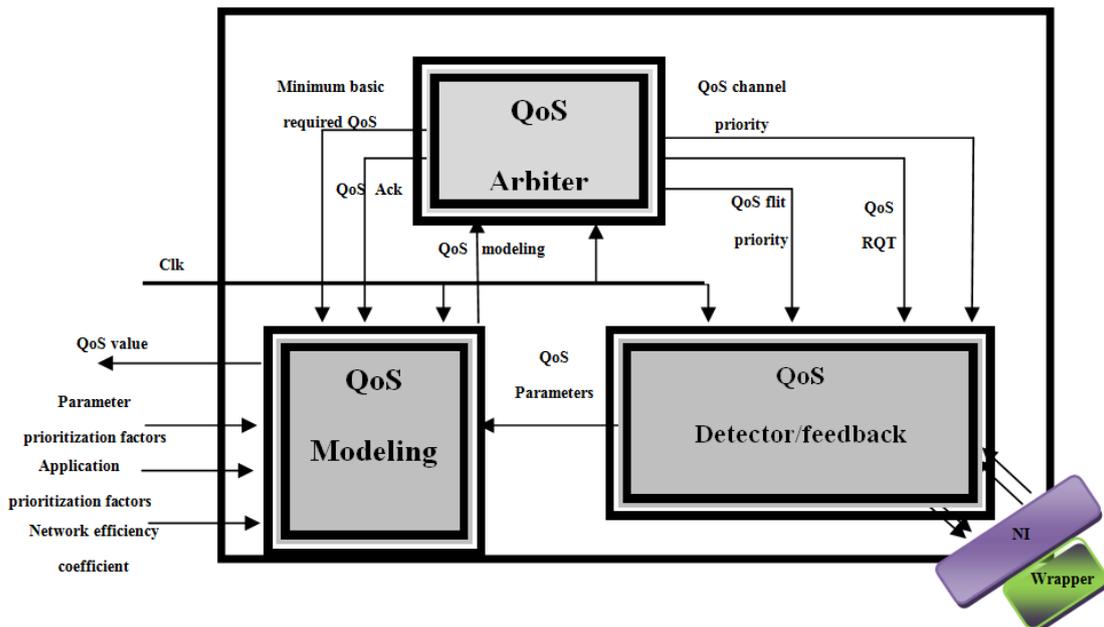


Figure 7. QoS module

In Figure 7 is shown the block diagram of QoS IP, including many functional blocks. QoS detector/feedback block exchanges data with other IP via network interface and wrapper as driver bloc. It provides many normalized parameters to QoS modeling.

To efficiently exploit QoS module features, QoS arbiter, the main important module, interacted with other routers to ensure flits flow and QoS modeling to provide a QoS value.

3.3. Adaptive Routing NoC Environment

The adaptive routing, used in this study, uses paths used to route new traffic between origins and destinations change occasionally in response to congestion.

The idea here is that congestion can build up in some part of the network due to changes in the statistics of the input traffic load. Then, the routing algorithm should try to change its routes and guide traffic around the point of congestion [26].

Our strategy selects randomly horizontal and/or vertical broking links scenarios on the entire NoC. This forces the system to search a new path between sources and destination.

The algorithm is given in pseudocode and is defined below, where t is the iteration time of simulation, while duration time is the total simulation time.

Algorithm: dynamic routing scenarios.

1. Read inputs data
 2. **for** all iterations i **do**
 3. Read new NoC state,
 4. **if** $t \leq$ Duration time **then**
 5. Compute shortest path
 6. Ap_i sends data
 7. **if** NoC state changes **then**
 8. Recompute the new shortest path
 9. **end if**
 10. **else**
 11. Compute QoS parameters
 12. **end if**
 13. **end for.**
-

Our routing algorithms are based on the notion of a shortest path between two nodes. It is based on distance vector (DV) algorithm for determining the shortest path among the available links by using the code (\$ns rproto DV). We use the specific code (\$ns rtmodel-at [random time] [down/up] [random {node_id} random {node_id + 1}]).

The script should calculate the QoS parameters such as EED, Thp and PLR using an AWK language.

4. QoS Metrics Modeling Requirements

4.1. QoS Modeling

As explained in [22,24], In a multi-application environment (ap_1, ap_2, \dots, ap_m), we define for each application ap_i a set of parameters ($p_{i1}, p_{i2}, p_{i3}, \dots, p_{in}$). Many parameters influenced directly QoS like Thp and others inversely as EED and PLR, so we set:

$$\sigma(p_{ij}, ap_i) = \begin{cases} 1 & \text{if QoS is proportional to } p_{ij} \\ -1 & \text{if QoS is inversely proportional to } p_{ij} \end{cases} \quad (1)$$

QoS performance parameters should be normalized as \hat{p}_{ij} , with: $p_{ijmax} = \text{Max}\{p_{ij}^{\sigma(p_{ij}, ap_i)}\}$

$$\text{And } p_{ijmin} = \text{Min}\left\{p_{ij}^{\sigma(p_{ij}, ap_i)}\right\} \quad [16].$$

Then:

- a. For increasing parameters when application value increases:

$$\hat{p}_{ij} = \frac{p_{ij}^{\sigma(p_{ij}, ap_i)} - p_{ijmin}}{k * p_{ijmax} - p_{ijmin}} \quad (2)$$

- b. For decreasing parameters when application value increases:

$$\hat{p}_{ij} = \frac{p_{ijmax} - p_{ij}^{\sigma(p_{ij}, ap_i)}}{k * p_{ijmax} - p_{ijmin}} \quad (3)$$

- c. $k \geq 1$: represents the network efficiency coefficient.

If we suppose that we have m applications, QoS can be expressed by the following model:

$$apm = \alpha m1 * \hat{p}1m + \alpha m2 * \hat{p}2m + \dots + \alpha mn * \hat{p}nm. \quad (4)$$

Then:

$$QoS = QoS_0 + \beta 1 * ap1 + \beta 2 * ap2 + \dots + \beta m * apm. \quad (5)$$

Referring to the proposed model in (5), QoS can be presented by the following formula:

$$QoS = QoS_0 + tr \left[\begin{matrix} diag(\beta_i)_{1 \leq i \leq m} * (\alpha_{ij})_{1 \leq i \leq m} * (\hat{p}_{ji})_{1 \leq i \leq m} \\ 1 \leq j \leq n & 1 \leq j \leq n \end{matrix} \right] \quad (6)$$

To evaluate QoS, we chose the parameter prioritization factor α_{ij} , the application prioritization factor β_i and the minimum acceptable value QoS_0 .

Where α_{ij} and β_i are arbitrarily fixed referring to the following equation:

$$\sum_{j=1}^n (\alpha_{ij}) = 1 \text{ and } \sum_{j=1}^n (\beta_j) = 1. \quad (7)$$

4.2. Computing QoS criteria Weights

It is hard to accurately quantify the weight of each QoS criterion in the case of decision situations involving subjective judgments. Multiple criteria decision making method is most applicable to solving problems that are characterized as a choice among alternatives. Furthermore, Analytic Hierarchy Process (AHP), is used to facilitate decisions that involve multiple competing criteria and to determine the numeric weights of the QoS parameters [11,27].

In this context, we present our proposed approach of QoS metrics modeling for NoC by decomposing the problem into an AHP hierarchy as shown in Figure 8:

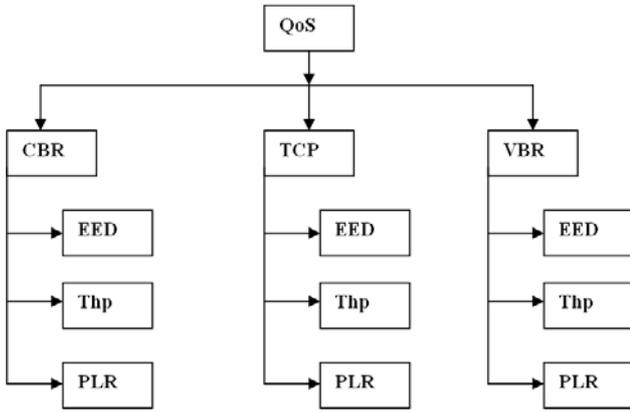


Figure 8. QoS metric hierarchy

End to End QoS Metric modeling employs a range of measurable performance metrics such as Thp, EED and PLR. Therefore we gave the PLR the value of 4. However Thp was the most important factor followed by EED so we gave them the values 6 and 5 respectively. Table 1 is then normalized using AHP Process, in order to obtain the weighted eigenvector for the comparison matrix for QoS parameters in Table 3.

We shall look all the comparison matrices required in a pair-wise manner (parameters versus parameters) as follows:

Table 1. QoS parameters Criteria Comparison Matrix

proposed scale	5	6	4
parameters	EED	Thp	PLR
EED	1	0.83	1.25
Thp	1.2	1	1.5
PLR	0.8	0.67	1
Total	3	2.5	3.75

Table 2. Normalized Comparison Matrix for QoS parameters

	EED	Thp	PLR
EED	0.33	0.33	0.33
Thp	0.4	0.4	0.4
PLR	0.27	0.27	0.27

Table 3. Weighted Eigenvector for Comparison Matrix for QoS parameters

EED	0.33
Thp	0.4
PLR	0.27

Table 4. Network Comparison Matrix for QoS application

proposed scale	2	5	6
	CBR	TCP	VBR
CBR	1	0.4	0.33
TCP	2.5	1	0.83
VBR	3	1.2	1
Total	6.5	2.6	2.16

Within the same context, we gave CBR, TCP and VBR the values 2, 6 and 5 respectively with the aim to obtain

the weighted eigenvector for the comparison matrix for QoS applications in Table 6.

Table 5. Normalized Network Comparison Matrix for QoS application

	CBR	TCP	VBR
CBR	0.154	0.154	0.154
TCP	0.385	0.385	0.385
VBR	0.461	0.461	0.461

Table 6. Weighted Eigenvector for Network Comparison Matrix for QoS application

CBR	0.154
TCP	0.385
VBR	0.461

5. Experimentation, Results and Analysis

We used [28] to create a significant scenario in which it was possible to test the network performance of the approach previously described.

All links study have a capacity of 100 Mbps. We use the SFQ algorithm (Stochastic Fairness Queuing), which is supposed to be an equitable distribution algorithm. The communication of three applications starts simultaneously in the same time using for each, three parameters are used to measure the QoS of an IP connection and quantify end-to-end NoC performance. NoC should ensure the negotiated QoS by satisfying certain values of these parameters.

5.1. End-to-end Delay

In the light of previous simulations, we can conclude that contrary to CBR and TCP, VBR is the application that gives better results to network performances. Also, with increasing packet size, EED average increases for all applications. So, EED is affected by flows sharing the same links, since each link capacity is divided among all applications sharing the link. Whereas, the shortest path used in dynamic routing scenario has a better end-to-end delay.

5.2. Throughput

It is clear that if too many packets try to access routers, it become overloaded with too much data and the input queue is full, resulting in a set of packets being dropped which affects the throughput of all applications. NoC links are shared by applications, and when the offered load is excessive, a portion will be rejected and Thp will be function of the difference between offered and rejected load.

In fact, Figure 10 depicts the variation of the concerned metric Thp average with the available packet size for CBR, VBR and TCP applications. We observe that CBR traffic data rate do not fluctuate during transmission. While VBR gives the best Thp average variation, TCP gives worst results.

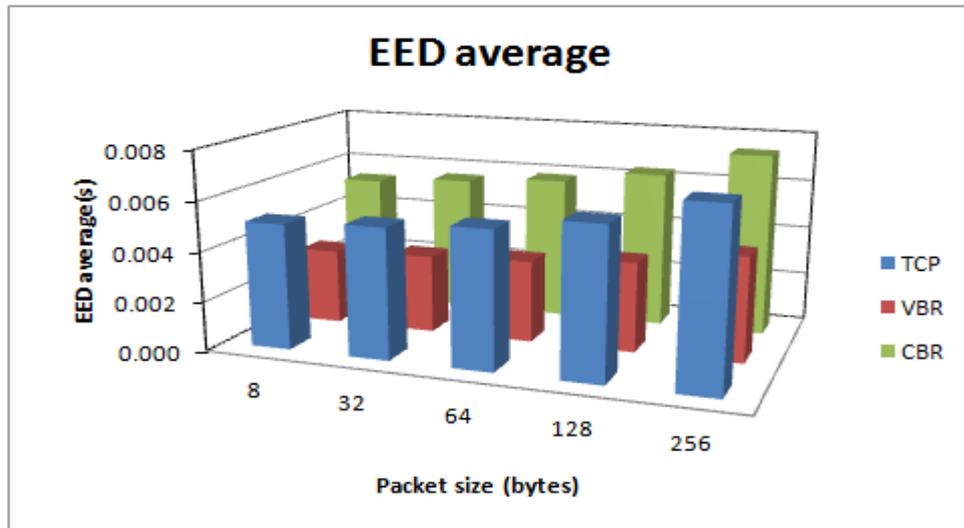


Figure 9. EED average of three applications according to packet size

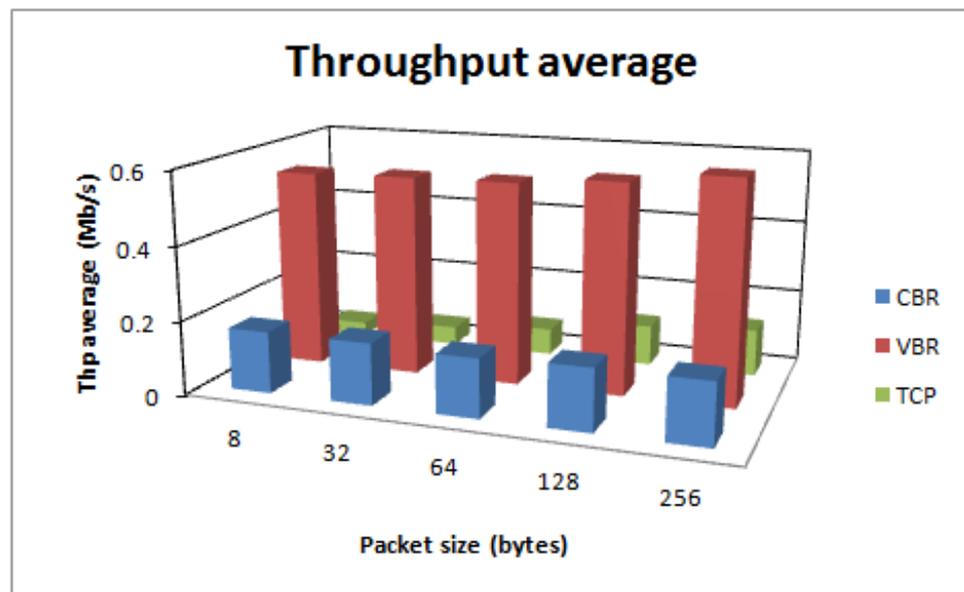


Figure 10. Thp average of three applications according to packet size

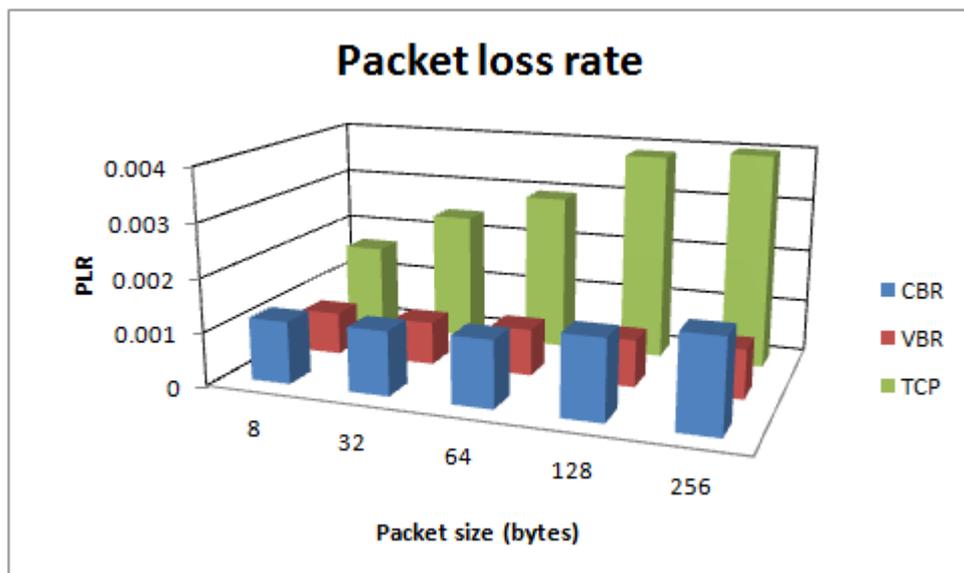


Figure 11. Rate of packet loss of three applications according to packet size

5.3. Rate of Packet Loss

When a flit enters the router, the router attempts to forward it to output ports. If it cannot find an appropriate route, flit is queued in the input queue of the incoming interface to be processed. Which has an impact on the EED and PLR. In fact, packet loss is mainly due to congestion in path, however, in case of link failure, packet can be lost due to time out [29,30].

The graph in the Figure 11 shows the variation of the packet loss rate as a function of the available packet size for CBR, VBR and TCP applications. PLR is measured as a percentage of packets lost with respect to packets sent.

As shown in Figure 11, TCP gives the highest rate of packet loss variation. When packet size is taken 8 bytes then packet loss is 0.15%, at packet size 64 bytes packet loss is 0.22% and so on. So it can be seen that packet loss rate is continuously increased as the packet size is increased. Otherwise, it increases its latency due to additional time needed for retransmission. Also, TCP flows reduce UDP Rate of packet loss to some technical and physical constraints, and prevent efficient use of bandwidth for large data transfers.

We can also observe that the PLR variations of CBR and VBR increases with the increase of packet size.

Furthermore, Packet loss can reduce throughput for a given application. In order to remedy this problem, QoS-IP module fairly splits available bandwidth between multiple applications flows when a given router or NoC link reaches nears its maximum capacity.

5.4. QoS Measurements and Analysis

As mentioned before, QoS parameter is the instance to represent the quality of service to customers. In fact, we consider three QoS performance parameters, such as EED as π_1 , Thp as π_2 and PLR as π_3 , for three concurrent applications CBR ($i = 1$), VBR ($i = 2$) and TCP ($i = 3$) for different available packet sizes.

Figure 11 shows the percent QoS in relation to the packet size, the scheduling techniques, and parameter and application prioritization factors, however we chose $QoS_0 = 10\%$ of the value of the ideal QoS.

Based on the QoS model indicated above and the equations described in [24], the QoS-IP module determines the values of normalized parameters, then the matrices of applications to finally determine the QoS values for each values of packet size. It provides a quantifiable representation of the overall QoS in NoC which was a very challenging task.

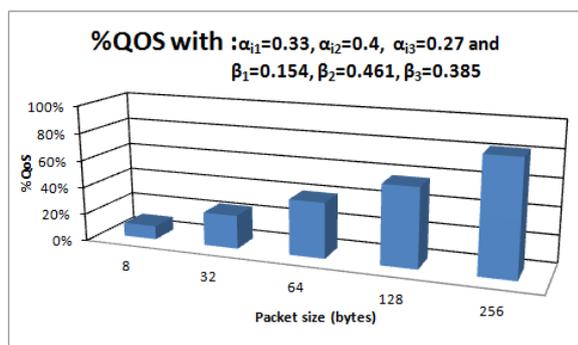


Figure 12. %QoS with parameter prioritization factors of three applications according to packet size

As shown in Figure 12, for example, when packet size is taken 32 bytes then %QoS is 19%, at packet size 128 bytes packet loss is 54% and so on. QoS of all application flows begin by 10% as a minimum basic required QoS value rate and has a highest value rate of 78% due to parameter prioritization factors.

QoS values, as indicated by the network performance results, is proportional to the values of packet size.

Whether it succeeded in its purpose would depend, obviously, on the AHP method, where prioritization factors have an impact on the QoS values. However, by varying prioritization factors, the QoS model approach gives for each packet size a QoS value and helps to make up the efficiency of the QoS metrics evaluation.

Moreover, by examining the approach proposed in [23], which is based on the deficit weighted round robin technique for the management of the data queuing, it has shown that approach used two parameters for one application service. Whereas the overall improvement in our extended approach is the combination of many QoS parameters of many applications building in a QoS model.

6. Conclusions and Perspectives

We developed a QoS-IP module in NoC architecture and demonstrated its performance enhancement over the previous work. The communication system became self-regulating. It can produce stability and reduce the effect of NoC communication problems and the gap between the measurement and the required end-to-end QoS-NoC.

We have also demonstrated that dynamic routing improves better performance and reduces congestion by choosing more optimal links.

Although the QoS is qualitative and not measurable, we have proposed, in this work, a new approach of its quantifiable representation using AHP method on multi-parameter and multi-application environment.

Therefore, further research should be done in order to find additional simplifications and improvements in the approach.

References

- [1] Benini, L., D. M., G. «Networks on chips: A new SOC paradigm», IEEE Computer, 35 (1), (2002), p.70-78.
- [2] Benini, L., D. M., G. «Networks on chips: Technology and tools», San Francisco, CA: Morgan Kaufmann Publishers. (2006).
- [3] Bolotin, E., C. I., G. R., K. A. «QNoC: QoS architecture and design process for network on chip», Journal of Systems Architecture, Special issue on Networks on Chip, 50(2-3), (2004), p.105-128.
- [4] Goossens, K., D J., G. O. P., G.P. S., R., A., R. E. «A design flow for application-specific networks on chip with guaranteed performance to accelerate SOC design and verification», In Proceedings of the design, automation and test in Europe conference and exhibition (DATE'05), (2005), p.1182-1187.
- [5] Rijpkema, E., Gooss. K., Radu., A. «Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip», In Design, automation and test in Europe (DATE'03), (2003), p.350-355.
- [6] Rijpkema, E., Gooss. K., Wiel. P. «A router architecture for networks on silicon», In The 2nd workshop on embedded systems (PROGRESS'2001), (2001), p.181-188, Veldhoven.
- [7] Bjerregaard, T., M. S. «A survey of research and practices of network-on-chip», ACM Computing Surveys, 38, (2006), p.71-121.

- [8] Chunlin, L., L. L. «Cross-layer optimization policy for QoS scheduling in computational grid», *Journal of Network and Computer Applications*, Elsevier, p.31, (2008), p.258-284.
- [9] Plestys, R., Vili. G., Sand. D., Vask. D., Kavaliunas, R., Kaunas, R. «The measurement of grid QoS parameters», In *IEEE, Proceedings of the ITI 2007, 29th international conference on information technology interfaces*, (2007), p.703-707.
- [10] Zhiang, W., & J. L. «The measurement model of grid QoS», In *IEEE, 10th international conference on computer supported cooperative work in design*, 2006, (2006), p.1-6, Nanjing.
- [11] Saaty, Thomas L., «Decision making with the analytic hierarchy process», *Int. J. Services Sciences*, vol. 1, no. 1, (2008), pp. 83-98.
- [12] Beigne, E., C. F., V., P., C., A., R. M. «An asynchronous NOC architecture providing low latency service and its multi-level design framework», In *11th IEEE international symposium on asynchronous circuits and systems*, (2005), p.54-63.
- [13] Goossens, K., Dielissen J., R. A. «Ethereal network on chip: Concepts, architectures, and implementations», *IEEE Design and Test of Computers*, 22(5), (2005), p.414-421.
- [14] Marescaux, T., Cor. H. «Introducing the SuperGT network-on-chip; SuperGT QoS: More than just GT», In *Proceedings of design automation conference (DAC'07)*, (2007), p.116-121. San Diego, CA: ACM, IEEE.
- [15] Joven, J., M. A., Angiolini, F., Benini, L., D. M. G. «Exploring programming model-driven QoS support for NoC-based platforms», In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis (CODES/ISSS '10)* (2010), p.65-74, Scottsdale, AZ.
- [16] Yang, L., Huac. H. «Grid service selection using QoS model», In *Third international conference on semantics, knowledge and grid (SKG 2007)*, (2007), p.576-577.
- [17] Helali, A., Nasri Salem «Network on chip switch scheduling approach for QoS and hardware resources adaptation», *International Journal of Computer Sciences and Engineering Systems (IJCSES)*, 3(1), (2009), p.29-35.
- [18] Helali, A., S. A., Bhar, S., Nasri Salem «Study of network on chip resources allocation for QoS management», *Journal of Computer Science*, 2(10), (2006), p.770-774.
- [19] Helali, A., S. A., Nasri, S., Divoux, T. «An approach for end-to-end QoS and network resources management», *Computer Standards & Interfaces Journal*, 28, (2005), p.93-108.
- [20] Liu, M., Xu, S., Sun S. «An agent-assisted QoS-based routing algorithm for wireless sensor networks», *Journal of Network and Computer Applications*, Elsevier, 35(1), (2012), p.29-36.
- [21] Nasri, Salem «Network on chip: A new approach of QoS metric modelling based on calculus theory», *International Journal of Communications, Network and System Sciences*, 3(5), (2011), p.53-60.
- [22] Saadaoui Abdelkader, Nasri, Salem «NoC: QoS metrics modelling and analysis based on dynamic routing», *International Journal of Distributed and Parallel Systems (IJDPDS)*, 3(2), (2012), p.43-52.
- [23] Nasri, Salem «New approach of QoS metric modeling on network on chip», *International Journal of Communications, Network and System Sciences*, 4(5), (2011), p.351-355.
- [24] Saadaoui Abdelkader, Salem Nasri. «Extended QoS Modeling based on Multi-Application Environment in Network on Chip», *International Journal of Electronics*, 102(1), (2015), pp. 89-106.
- [25] Yunus, S. A. M. J., Marsono, M. N., & Ibrahim, I. «Modeling router hotspots on network-on-chip», In *IEEE, advanced communication technology (ICACT), 13th international conference*, (2011), p.896-900, Seoul. IEEE.
- [26] Dally, W. J., A. H. «Deadlock-free adaptive routing in multicomputer networks using virtual channels», *IEEE Transactions Parallel Distributed Systems*, 4(4), (1993), p.466-475.
- [27] Nachiappan, C., Mis., A. K., Kan. M. T., Siv. A., Mutlu, O., Das, C. R. «Application-aware prefetch prioritization in on-chip networks», In *Proceedings of the 21st international conference on parallel architectures and compilation techniques*, (2012), p.441-442, Minneapolis, MN.
- [28] NS-2. The network simulator NS-2: <http://www.isi.edu/nsnam/ns>, 2007.
- [29] Winter, M., Pru. S., Gerh. P. F. «Hierarchical routing architectures in clustered 2D-mesh networks-on-chip», In *IEEE, SoC design conference (ISOCC)*, (2010), p.388-391.
- [30] Yiping, D., Z. L., Wata., T. «An efficient hardware routing algorithms for NoC», In *Conference on Tencon 2010 IEEE region 10*, (2010), p.1525-1530, Fukuoka. IEEE.