

NOC Architecture Comparison with Network Simulator NS2

Neila MOUSSA, Farah Nasri, Rached Tourki

Monastir University

Abstract— The current density integration in integrated circuits can have quite complex SoCs. Problems of interconnections between IP blocks become a critical point that current communication structures are no longer able to solve. These problems are mainly related to the notion of QoS becomes an important concept ensuring maximum reliability. Thus, solutions based NoC offer good prospects to overcome the current limitations of particular topologies Bus. NoC then constitutes a new paradigm for interconnecting SoCs. Through this paper, we tested four NoC topologies in different simulations with NS-2. Through metrics evaluated we were able to identify a selective classification of these topologies.

Network on Chip (NOC) and Network Simulator (NS2)

I. INTRODUCTION

SoCs (System On Chip) become increasingly obsolete share life reduced products "general public". Products must adapt to a large number of standards (Wi-Fi, WLAN, GPRS ...). The main trouble faced by designers of SoCs will provide reliable operation, proper functionality and interactive components. The effectiveness of interconnection has become one of the key points to the success of the design of SoCs. The interconnection networks have been studied for several decades. This is the case of computer networks, telephone networks [1]. However the last ten years we have seen a rapid evolution of the interconnection of systems on chip technology. In particular, the global telecommunications conference in 1992 [2] proposed to replace the bus by networks on chip interconnections based routers or NoCs that are likely to provide effective solutions to complex integration of systems chip [3] [68].

New applications in nanotechnology today require more complex networks and increase their performance. In other words, it is important to introduce the concept of quality of service in networks on chip. In practice, particularly in the industrial context, simulation remains predominantly used as a method of validation of NoCs during the design phase. Even if the simulation time is not negligible and can be very long this limits opportunities for architectural exploration.

In our implementation, the simulation as an evaluation of a set of QoS parameters method remains a strategic choice during the design phase of integrated circuits (eg VHDL programming that addresses the more physical behavior and transmission).

To evaluate the transmission in a network on chip using different topologies must have an idea about all the requirements of feasibility. For that several simulators have been developed and depending on the particularities of each simulation tool, we have chosen the most appropriate simulator. To avoid problems with installation and configuration, we had to choose a stable simulator sufficiently tested and the most suitable for our work environment (simulation data transmission in a network on chip using different topologies). These conditions are verified by NS-2 simulator is extremely flexible. It allows the study of difficult cases to replicate in reality. It provides the flexibility to easily change a plurality of parameters of the network (which is not the case when simulating on a real network). So compared to the experimentation on a real network, the NS-2 tool saves time and money. The latter is certainly the network simulator used most. Due to its popularity, many additions are available to him as 802.11, Evalvid, etc..

In this paper, we first clarify some places NS2 simulator features selected. Subsequently we will describe the steps taken to prepare the simulation environment. Finally, we turn to the simulation portion to define different QoS parameters evaluated and interpret the results.

II- Network Simulator NS2

Like any simulation tool, NS-2 allows the study of the existing design, validation and evaluation of performance and this in the field of network protocols [66] mechanisms. Now, NS-2 has limitations like any simulator. Among which was its inability to simulate large networks and consumer equipment [67] level (processor, memory, large trace files, ...), which makes the simulation time long enough. In fact, working on such a simulator you have to learn its language and script otel have an idea about its inner workings which relies on a certain degree of complexity compared to other simulators. For a beginner researcher, it takes about two years to complete [61] control.

a. Modeling and implementation of NoC in NS-2

In this study, we sought to characterize the on-chip network in NS-2 through four different topologies. The results are presented as curves, resources (nodes) use random traffic generators. Different statistical distributions are possible to change the temporal distribution of packages, their sizes and locations. The simplest model is an equitable distribution of packets, which is a network where all resources functions (sending and receiving data), sizes and distributions because it is equivalent to model the operation of communication network and independently of a high processing units that will exploit the network level.

b. Parameters of the NoC

NoC implemented in NS-2 is composed of routers and resources, routers are connected in pairs and each router is connected to a resource through RNI (network interface resource) which covers the first four layers of the communication model suitable for the NoC. All resources can be heterogeneous (either a

DSP, CPU, memory, ...) and they have the same weight in this context.

Communication is between resources through routers. We have implemented four NoC topologies in NS-2: 2D mesh, ring, octagon and the torus. Each topology can be expandable to an unlimited number of blocks, but given the limitations we have chosen as the dimension $k = 4$ and $d = 2$ in each model. With a total equal router k^d and the number $k^d(d-1)$ of equal resources.

The links between these blocks are duplex type, with a maximum transmission rate of 200Mb / s and a delay of 0.1ms using UDP as exchange protocol. It provides a non-guaranteed data delivery because it does not require an acknowledgment when sending, so it is charging less (during simulation) than the TCP. With these parameters the total traffic can be achieved, according to this equation, the value of 6 Gb / s.

$$\text{Bande passante totale} = 2 k^d b \quad ; \text{avec } b = 200 \text{ Mb / s}$$

In NS-2, the algorithm used is the default routing "Dijkstra SPF". [62] It uses the adjacency matrix and link costs in the topology. It is executed once before the start of the simulation, it is considered static. Unlike dynamic routing, it does not require substantial hardware resources and high performance processor.

Buffer FIFO is selected, it is used in most existing architectures routers to temporarily keep the received packets. With NS-2 configuration buffer size is fixed at eight in the entrance of each router and reset for output. We used the mechanism "DropTail" in NS-2 to support momentarily overload, where the last packet arrived is automatically rejected [62], so it is similar to FIFO and provides simple management of queue. While the simulation is made with a correspondingly high level tool NS-2 regardless of a specific type of application, then the traffic generation is selected in an arbitrary manner wherein the IP can be selected as sources or destinations to ensure that the communication is in all blocks. We set the size of packets circulating in the network to 64 bits. This table summarizes the parameters used for the simulation:

NoC	
Paramètre	Valeur
Topology	mesh 2d / anneau / torus / octogone
Number de nodes	16
Number de routers	16
Connexion	router-router / routeur-ressource
Highest throughput of connexion	200 Mb/s
Delay maximal des liens	0.1 ms
Type of FiFO	FIFO (DropTail)
FiFO Size	8 paquets
Data size	64 Bits
routing Type	Statique SPF
Traffic behaviour	exponential traffic generation
Transmission protocol	UDP

Tab1. Communication parameters for the NoC

c. Description of used scenarios

Exchange of video data

For video traffic, we chose the first resource connected in each topology as a receiver of the data. Each wireless node sends to the base station a movie cut in 1000 bytes. The playback speed of the video is 30 frames per second with 176x144 resolution. The base station is responsible for routing the data to the IPs of the NoC. The duration of our simulation is defined as the total completion of the last video packet sent from the source to the reception early costs.

$$T_s = \frac{\frac{T}{fps} * (F + 1)}{1000}$$

Simulation time T_s depends on size of cut T packets are divided on the number of frames per second (FPS) video sequence, while it is multiplied by the total number of received image F by adding 1. Time is calculated in seconds. If sending multiple sequences, the value of F will be the largest number of existing images.

Exchange internal data in the NoC

As we already mentioned, NS-2 is a high-level simulator based on event management, why was defined six different behaviors of local traffic circulating in the NoC:

- Scenario 1: The first resource sends packets to the rest of the resources.
- Scenario 2: The first seven resources send packets to the last seven.
- Scenario 3: Send random between resources.
- Scenario 4: The last seven resources send packets to the first seven.
- Scenario 5: All the resources to send packets to the first resource.
- Scenario 6: Packages are sent to the mobile nodes from the first resource.

The distribution of these different scenarios during the period of the simulation (which depends on T_s) is described in Fig.1

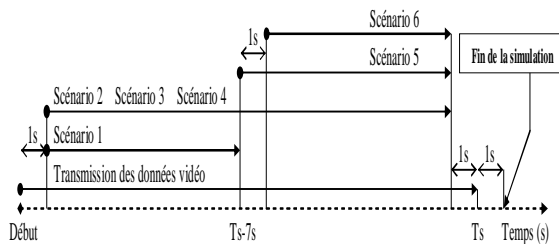


Fig.1. Distribution of scenarios over time

These events will be implemented in each topology (2D mesh, ring, octagon, torus), so the traffic used in NS-2 is the exponential [62] which is defined by four parameters:

PktSize_: packet size generated.
Burst_time_: the deadline for transmission activity.
Idle_time_: the rest period of transmission.
Rate_: frequency of sending in the "burst_time" period
 For all these parameters, we only changed the packet size to 64 bits for the rest we left the default values already defined in the NS-2 simulator. This traffic uses UDP agent as a transport agent for the network. We do not exactly defines the structure of messages circulating in the network, we adopted then existing general encapsulation.

In our simulations we used two video sequences [63]: "foreman_qcif.yuv" in QCIF(Quarter Common Intermediate Format) format with 176x144 resolution with 400 images and "st_stefan_cif" in CIF format with 352x228 resolution with 90 frames.

1.2.3. The residue network

Triggering event stop "stop" in NS-2 for all existing traffic generators in the simulation will stop the sending packets. But it is possible that there are packages that are en route to their destinations in the network, they are called the "residual network". It is obvious that it is inevitable at the end of each simulation and the simulator will spend time to clean the entire network.

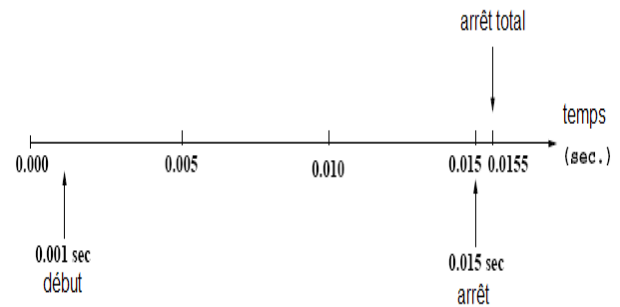


Fig.2: Timeline for a process simulation

Fig.2 A through 4, one distinguishes a cleaning time of 0.0005 s for a flow rate of 200Mb / s, then the time depends on bandwidth and it has no relationship with the total simulation time calculated relative the transmission time video traffic [29]. To this we have added a range of additional 1s at the end of the scenario.

d. Handling code simulation

Before running a simulation, we introduced the parameters in NS-2 through TCL scripts. The parameters consist including: number of node links, topologies, scenarios (different events that will occur), protocols implemented, etc..

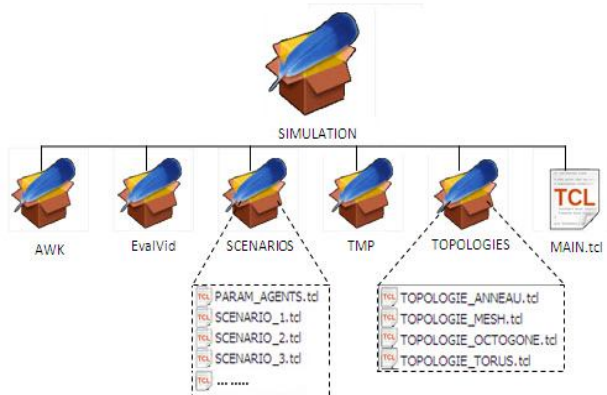


Fig.3: Tree of the project

Our simulation project is designed in a way to allow extensibility at any time, in fact scenarios and topologies are each implemented in a separate file TCL what makes our project quite understandable and easy to change (Figure 1 of 2). In addition, different topologies are designed so as to allow easy change of the size which reflects the dynamic. The "main" function makes calls from other codes (Fig 3).

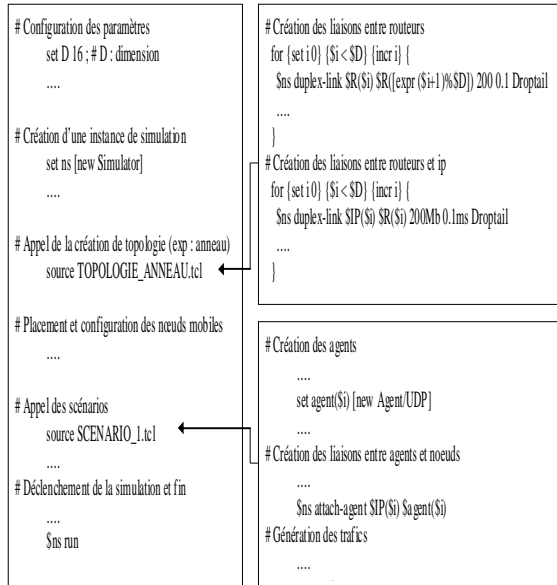


Fig4. Structure of Appeal in the code of the simulation

The outputs of the simulations are text files called trace file. These files are structured inputs. Each entry corresponds to an action performed by a node (send packet, packet rejection, setting queue, receiving a packet).

These files must be processed to calculate the values of bandwidth, loss, delay and jitter to the scenarios in question. For example, to calculate the rate of loss, just count the entries starting with "d" (drop = rejected) and the entries starting with "r" (received = received), thus extracting the loss rate. This processing is performed by the processing language AWK files. Finally, tracing curves is performed by Gnuplot. In addition, the simulator allows the creation of an animation file to visualize the simulation on the NAM GUI.

e. Methodologies for the analysis of the results
Recording simulation and monitoring the queue

This is an integration value in each interval of time, which is defined in the code. For the study, we chose to use a "Record" procedure in which a loop recalculates the values of bandwidth and load queue will be called periodically to each link between resources.

```
proc record {} {
    ...
    # cycle d'appel de la procédure record
    set time 0.05
    set x 0
    foreach {index value} [array get qmon] {
        # mesure du nombre des paquets dans le file d'attente
        set number_of_packet [eval $value set parrivals_]
        # mesure des tailles des paquets dans le file d'attente
        set packets [eval $value set barrivals_]
        # pris du temps courant
        set now [$ns now]
        # remplissage des fichiers traces
        puts $Q($x) "$now $number_of_packet"
        if {$now != 0} {
            puts $B($x) "$now [expr {$packets*8}/(1000000*$now)]"
        }
    }
    incr x
}
# re-appel de la procedure
Sns at [expr $now+$time] "record"
```

Fig5. Recording Method "procedure Record" data

Therefore, this procedure is used to control how many packets arrive at the destination in every 0.05 seconds and also to identify the rate of incoming packets in the queue for the same time interval. To monitor the queue we are talking about the "QueueMonitor" class or monitoring the queue which is considered the most important method that facilitates the extraction of simulation results.

```
# surveillance sur les files d'attente dans l'entrée de chaque routeur
set qmon(tr$[expr $i+1]) [$ns monitor-queue $routeur($i) $routeur([expr $i+1]) ""]
# surveillance sur la file d'attente d'entrée de chaque routeur liée à une ressource
set qmon(ip_routeur$[i]) [$ns monitor-queue $ip($i) $routeur($i) ""]
```

Fig6. Example of a monitoring queue resources

According to the example in Figure 3 8 must be set at the beginning, monitoring each link created between the different resources to periodically monitor the status of each of these links according to several parameters.

The "QueueMonitor" class NS-2 contains many parameters gives various traffic information and the tendency of each bit and also each packet. These parameters can be identified, for example:

- npkts_: Number of packets in the queue.
- nlost_: Number Dropped packets from the queue.
- pdepartures_: Number of outgoing packets from the queue.

III. CURVES INTERPRETATIONS

In this section, we describe the results of simulations by setting quality of service target. We present these results as follows: for each parameter (bandwidth, delay, jitter and packet loss), we superimpose the curves of each topology studied (Appendix A: Maille

Ring, Torus, Octagonal) to differentiate between them and get away with relevant interpretations. To ensure the relevance, we adopted for our simulations the following assumptions:

- Maintain the same scenario for all topologies.
- Maintain the same network settings (speed, latency, protocols, applications, ...)
- Maintain the same size for different topologies (16 IPs and 16 routers).

We extract our results through three alternative. Indeed, in a first step, we conducted the evaluation of NoC in the presence of video traffic. To better judge the results, we had to simulate other behaviors such as lack of video traffic and the case of an existing application (MPEG-2) to give more realism to our work.

a. The total bandwidth

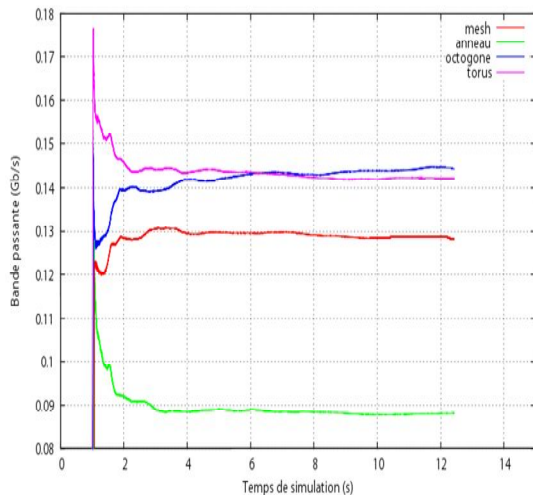


Fig.7 Bandwidth average for the four topologies

Figure 3 curve 12 represents the average bandwidth for the four topologies studied as a function of time. On a first glance, we see that these curves begin to 1s axis which corresponds to the start date of the scenario posed (in Figure 3 2). It can also distinguish between two levels of bandwidth for these curves. In fact, for the Octagon and Torus topologies, stabilization bandwidth occupies the interval [140-145] Mb / s and 130 Mb / s for 2D Maille. However, it does not exceed 90 Mb / s for the Ring. This stabilization persists despite the increase generated by the scenario 6 in the middle of traffic simulation. Confirming its not influence on the general traffic.

b. Average delay

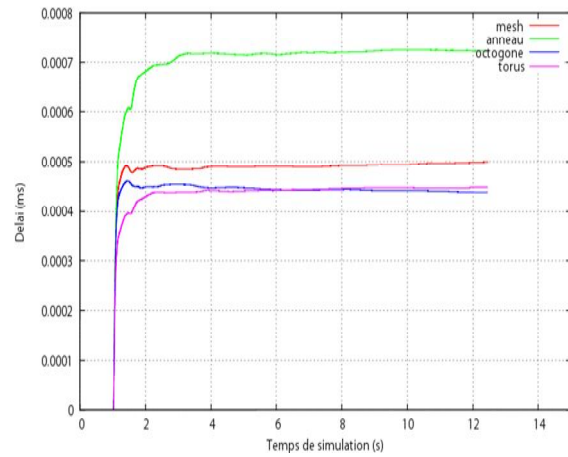


Fig.8: Delay for the four topologies

For QoS parameter also called latency period can be seen that all the curves have the same general shape (exponential rise followed by a steady pace). We can clearly distinguish the same grouping as the bandwidth: the Maille 2D Torus the Octagon and that approach at the stabilization value and the Ring that differs enough of them in the negative direction as to usual. Indeed, several milliseconds after the start of the simulation, and the Octagon Torus reach their stabilization with the value 0.45 microseconds, the 2D Maillé 0.5 microseconds and 0.7 microseconds with about ring which favors the first two topologies.

c. Average jitter

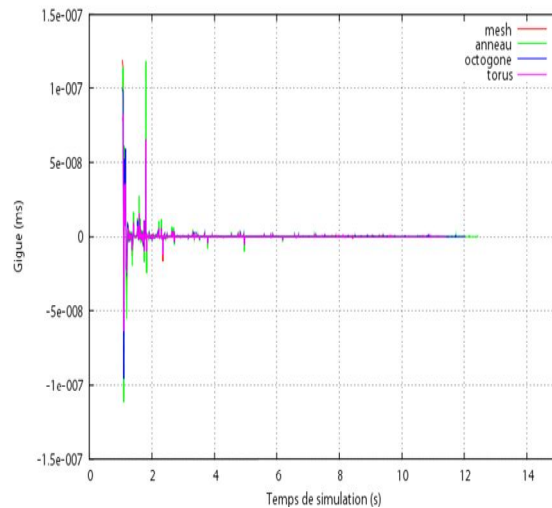


Fig.9: Average Jitter for the four topologies

In Figure 14 March disturbance jitter is clearly visible during the first two seconds, which corresponds to the launch scenarios. The presence of the jitter in the flow is from the abrupt change of the intensity due to

simultaneous traffic transmission trigger data. During the rest of the simulation, jitter is almost zero. For topologies Maille 2D Torus and Octagon curve terminals are close together. While the Ring topology is still different once other.

In the general case, jitter affects applications that transmit packets to a fixed rate and expect to receive the same rate (eg. Voice, video, etc) and this is our case with the video application.

d. The rate of loss

With a maximum throughput equal to 200 Mb links / s and a maximum equal to 0.1 ms (Table 3 2), we expect to have a good loss rate zero which corresponds to the result obtained for the four topologies . In fact, the maximum delay is measured in Figure 3 13 does not exceed 0.75 microseconds is quite low compared to the maximum delay link is of the order of 0.1 ms. This makes it impossible to fill queues, which suits the requirements of on-chip networks.

e. Load queue

Before starting the interpretation of this curve, we will outline the strategy we adopted to extract it. Indeed, we oversaw all queues put at the entrance of each router. The total number of entries varies from topology to another. We then selected to represent the results of supervision for the intermediate routers traversed by the video packets and their ends (in this case IP1 and BS in Appendix A). But given the congestion caused by the paces taken, we have chosen to present only the load queue of the link between the first router of each topology and resource.

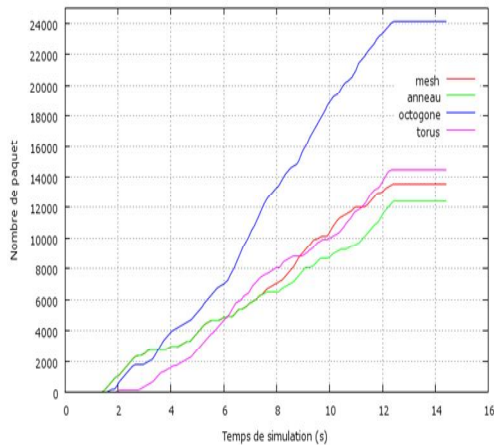


Fig 10. Burden of queue 1 ip link Router 1

According to this figure (Figure 3 15), we note that the router architecture octagon could pass 24,500 packets during the simulation. For others, they do not exceed 14500 packets.

f. PSNR

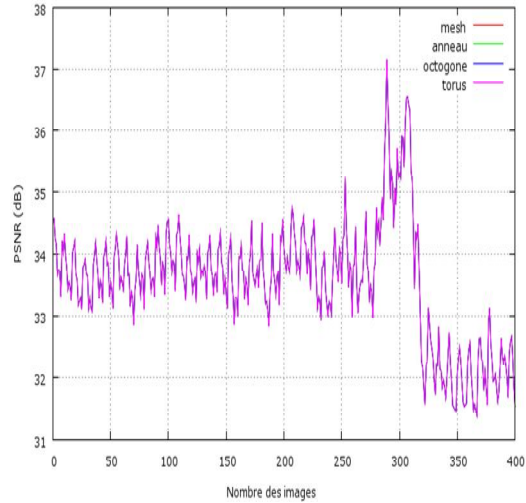


Fig11. PSNR of video flow

To assess the quality of the received video, we used the "EvalVid" tool in order to measure the PSNR. Indeed the curves in Figure 3 16 turns combined with an average value of 34 dB for the first 300 images. According to the equivalence PSNR-MOS, we can judge the quality of the received video is good (Table 2 1).

Through measurements in the presence of video traffic, we can take as a primary conclusion that the topology octagon favors to others. Indeed, it offers better bandwidth with a reasonable time and to increase the number of packets transmitted during the simulation. This topology can withstand a moderate increase in video resolution used. Following this part, to ensure more of our finding, we repeated the simulation by substituting video traffic data traffic.

IV. CONCLUSION

In this paper, we presented around different tools and languages used. In the second part, we specified topologies and scenarios used in our simulations. In the last part, we evaluated the QoS in the network on chip after four topologies (2D Maille, Ring, Octagon and Torus). Through the studied parameters (bandwidth, loss, delay, jitter, and PSNR), we compared these architectures to distinguish the one that offers the best results. The study is based on three simulation phases. Hence we concluded that the octagonal topology is preferable to other architectures.

REFERENCES

- [1] A. Grbic, S. Brown, S. Caranci, R. Grindley et al. « Design and implementation of the NUMAchine multiprocessor ». 35rd Design S.K. Tewksbury, M. Uppuluri, L.A. Hornak. « Interconnections/micro-networks for integrated microelectronics ». Global telecommunications conference. (GLOBECOM'92). 1992
- [2] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [3] M. Brière. « Flot de conception hiérarchique d'un système hétérogène -Prototypage virtuel d'un réseau d'interconnexion optique intégré ». Ecole Centrale de Lyon. 29 Novembre 2005
- [4] A. Hemani. « Network on chip: an architecture for billion transistor era ». IEEE NorChip Conference. Nov 2000
- [5] S. Riso, L. Torres, G. Sassatelli, M. Robert, F.Moraes. « Réseau d'interconnexion pour les systèmes sur puce : Le réseau HERMES ». Signaux, Circuits et Systèmes (SCS'04) Monastir, Tunisie. Mars 2004
- [6] M. Brière. « Flot de conception hiérarchique d'un système hétérogène -Prototypage virtuel d'un réseau d'interconnexion optique intégré ». Ecole Centrale de Lyon. 29 Novembre 2005
- [7] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [8] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.