SCPN evaluation of a Switched fabric CAN Network

Mohamed MAZOUZI† **Oussama KALLEL††** **Salem HASNAOUI†††** **Mohamed ABID††††**

Computer and Embedded Systems Laboratory-ENIS | Communication Systems Research Laboratory-ENIT | Communication Systems Research Laboratory-ENIT | Computer and Embedded Systems Laboratory-ENIS

Abstract

Nowadays distributed computing in complex embedded systems gain complexity when they are equipped with many microcontrollers which oversee many Electronic Control Units (ECU). High performance and predictability are the prerequisites for any large-scale networked system dependent on real-time data processing and analysis. Switched-fabric networks can provide fast and highly scalable hardware solutions and are now being increasingly used in distributed systems.

In this paper, we proposed a switched fabric CAN Network based on CAN Controllers and switch fabric. We have implemented in VHDL a CAN Controller and used an existing 8x8 Switch Fabric. We also evaluated the overall architecture using CPNTools. We separately have modeled the CAN Controller correspondently to its VHDL implementation and a SCPN model of the Switched Fabric CAN Network. Experiments are then applied to the model to extract information on network performance metrics.

Keywords: *Switched-fabric CAN network, CAN Controller, Switch fabric, VHDL, Performance evaluation, stochastic colored Petri nets, CPNTools.*

1. Introduction

During last decades the demand for sophisticated embedded systems requires the use of many connected equipments. A dedicated network bus [1] is used for connecting sensors, actuators in vehicles, robots and industries. Many serial buses were developed by car makers like MOST, J1850, SAE J1708, Byteflight, LIN... and CAN(controller Area Network). Most of them are specific to manufactures and not standardized [2].

CAN is one of the most popular fieldbuses [3,4]. More than 400 million nodes were sold worldwide. It is used in those applications that require fast and reliable communication [5].

Nowadays, more sophisticated buses are concurrent to CAN networks like FlexRay [6], recently appeared, and RTethernet. They offer higher speed to satisfy the high bandwidth required for modern vehicles, suitable for x-by-wire application. In contrast the usage of FlexRay [7] is not widely used due to its complex specification and high cost.

Current parallel bus-based [8] solutions present some problems. In fact, it's well known that the physical separation of cards is limited to usually less than 3 feet. There are also limited bandwidths, high protocol overhead and no deterministic performance. To reduce system life-cycle cost to ease system upgrades, the current solutions requires to re-architecture the software. The need for the availability (redundant buses, cards or entirely redundant systems) requires complex failover logistics and expensive hardware duplication. The use of switch fabric [9] provides new "net-centric" distributed systems that are much easier to scale, make fault-tolerant and upgrade. These high-performances distributed architectures promise to greatly simplify efforts and to add better capability and availability.

In our work, we realized this new architecture by its implementation in VHDL [10]. We also evaluated it using stochastic colored Petri nets. Our major contribution is the study of the performance of the proposed architecture to rise the lack of the current solutions. In fact, we demonstrated that CAN based Networks using Switch fabric have yet a well period before its replacement and CAN controller components manufacturers do not afraid for their production.

Our paper is organized as follow:

• The section 2 explains how switch fabric is necessary to

Manuscript received January 5, 2010
Manuscript revised January 20, 2010

interconnect high performance CAN Controller and details the proposed Switched fabric CAN Network.
• The section 3 focuses on the description of the CAN controller modules.
• Based on our modeling of the proposed architecture by stochastic and colored Petri nets (SCPN) [11, 12], we present, in the last section, some experiments which are applied to the model to prove that the proposed architecture has a good performance values.

2. The use of Switched fabric for High Perfor mance Distributed systems

A switched-fabric bus is unique in that it allows all CAN Controllers on a bus to logically interconnect with all CAN Controllers on the bus (Figure 1). Each node (CAN Controller) is physically connected to one or more switches. CAN Controllers may be connected to each other.

2.1 The proposed architecture

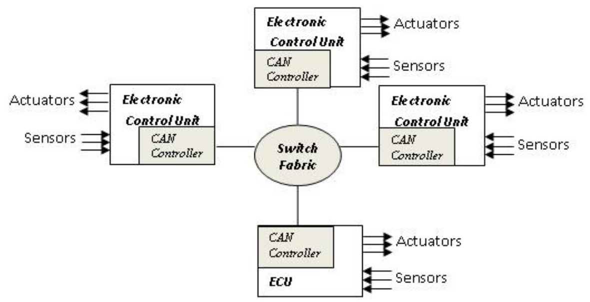

Fig. 1: Switched Fabric CAN Controller interconnect.

A key characteristic of switched fabric [13] is that they allow peer-to-peer communication between nodes without having to physically connect every node to every other node. Because a switched fabric CAN network employs switching to achieve logical connectivity and reconfigurability, these systems can be architected to be highly scalable. This topology results in a redundant network or fabric, in which there may be one or more redundant physical paths between any two nodes. A node may be logically connected to any other node via the switch. A logical path is temporary and can be reconfigured, or switched among the available physical connections.

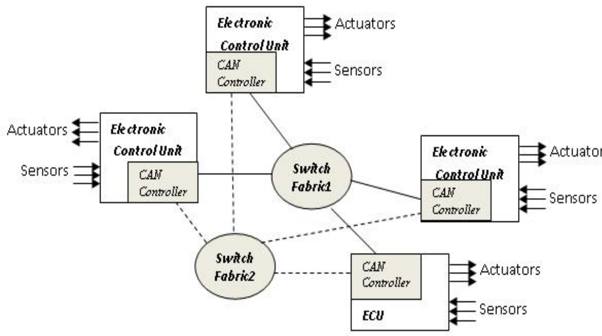

Fig. 2: Fault-tolerant Switched Fabric CAN Controller interconnect.

2.2 Important switching fabric modules

In this section, we briefly present the most important switching fabric modules architecture realized by John D. Pape [14]. The switching fabric is the physical connection within a switch between the input and output ports; it can be proved that all switches need a crossbar inside their switching fabric. Usually packets need to be queued in buffers when short-term overloading occurs, where the sum of input rates for a single output port exceeds the outgoing link rate. Buffering (queueing) characterizes all kinds of switches. We can use all types of queueing architectures: “Output Queueing (OQ)”, “Input Queueing (IQ)”, “and Combined Input OutputQueueing (CIOQ - Internal Speedup), Shared Buffer”, “Block Crosspoint Queueing and Crosspoint Queueing (CQ)”.
The following figure is being demonstrated on the 8 x 8 switch that connects 8 devices.

3. CAN controller Modules

As described at the beginning of this paper, we will detail the structure of the Controller adopted. The controller is divided in two main parts: the core and the interface. The core is responsible to ensure the CAN protocol specification while the interface is used to make easy the use of the core and standardized like a processor interface. These modules are implemented in VHDL language which describe the CAN protocol behavior. Figure 4 shows the controller structure, the core and the interface.

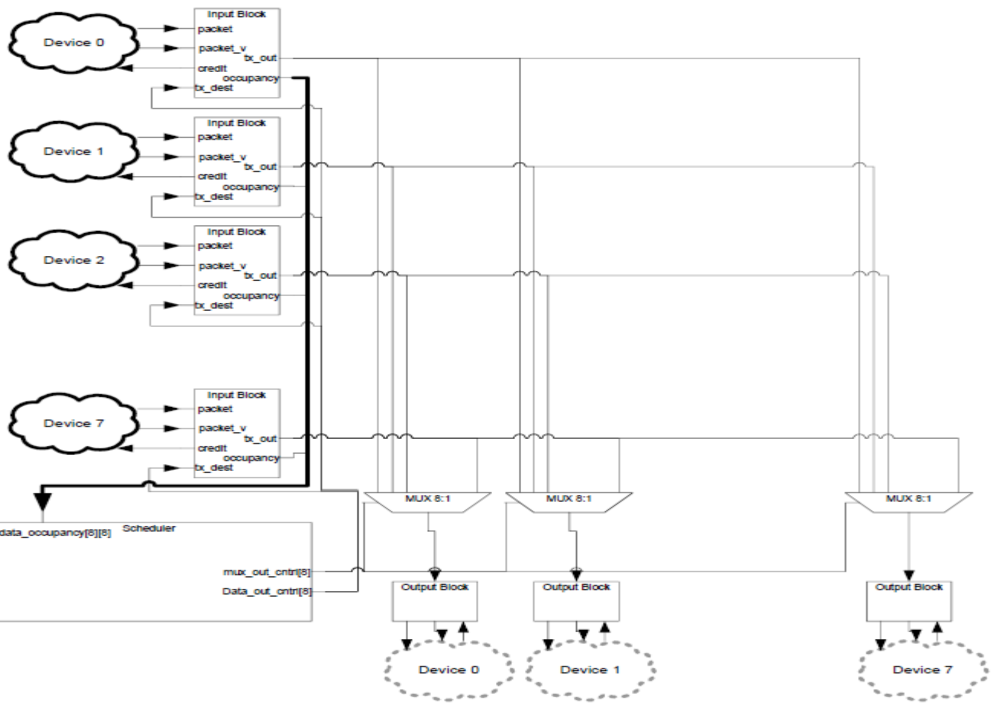

Fig. 3: Interconnect Implementation Block Diagram.

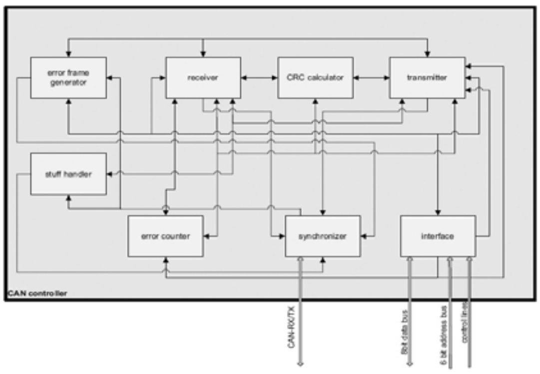

Fig. 4: CAN Controller module.

3.1 Receiver Module

The Receiver module is the main module of the CAN Controller. It's responsible for determining the state of the current frame (data frame) on the network: idle, SOF, ID...Most of the frame structures are decoded on this module. The input/output of this module interact with all modules. The figure 5 shows the I/O of this module.

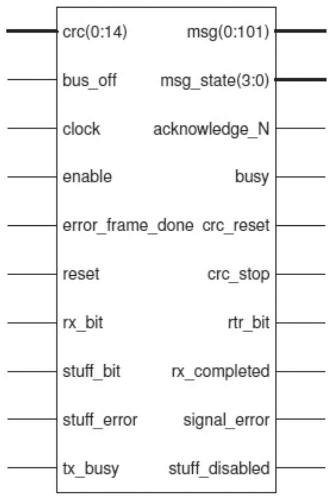

Fig. 5: CAN rx i/O.

3.2 Transmitter Module

The Transmitter module is intended to work in collaboration mainly with the Receiver module to distinguish if the node is actually in sending (data frame and remote frame) state or receiving state. The receiver dispatch the frame states to the transmitter, according to this and others signals from others modules it can decide whether is in OFF, SOF, TRANSMITTING, TX ERROR, REMOTE FRAME, REMOTE FRAME EXT or SUSPEND state. The input/output of this module interact with all modules as shown in figure 6.

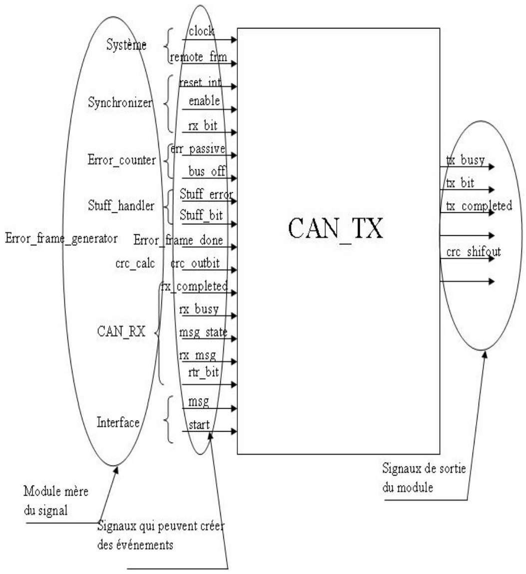

Fig. 6: CAN_TX.

3.3 Synchronizer Module

The Synchronizer module interacts with the Switched Fabric CAN Network via the transceiver to sample the bit stream and send the bit value (in case of transmitting). It's responsible for ensuring the specification bit timing by mean in which phase the bit segment is it: SYNC, P1RSJ, P1 …

The input/output of this module interact with all modules as shown in figure 7.

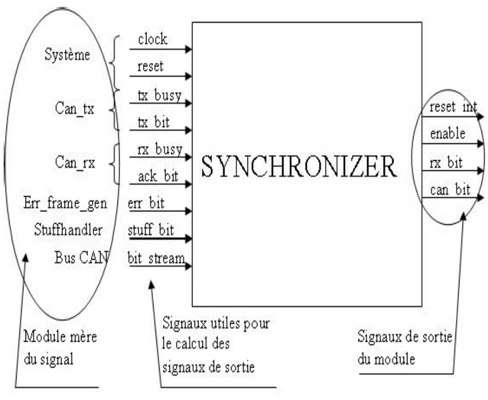

Fig. 7: Synchronizer IO.

4. SCPN Simulation and analysis

In this section, we give only some experiment results where the nodes are configurable by:

- The duration of data bursts (measured in clock cycles) to be transmitted (maximum is equal to Total_byte x 8x Bit_delay)
- The average delay between two successive requests for sending message.
- Message classes: Each node represents a class of message object. Node 0 is the class of high priority messages denoted by two dominant bits on the position ID0, ID1. The second node 1 is the class for medium priority messages denoted by Res, Dom or Dom, Res on the position ID0, ID1. The last node 2 is for low priority messages denoted by Res, Res value on ID0, ID1 bits.

We prepare another paper in which we give all the SCPN models of our architecture based on CAN Controllers and Switch fabrics. The results given below use these models. The throughput is the total messages sent divided by the total messages generated in the time spent [15].

Latency in this case corresponds to the average time used for winning the Network and to transfer data. Loss probability is the total message lost (due to full FIFO or higher message being transmitted when the actual message is being sent) divided by the total message requested.

In CPN tools there is a monitor tools which provide statistics measurement on either places or transitions [16].

CPNTools does not support non terminating simulation to investigate the steady state of the modeled system. However a group monitor is used to detect network stability. In fact such condition uses complicated algorithm not yet supported by the tool. Using the available monitor features of CPNTools we create conditions to know when the network is stable. The basic idea was to compare the last latency delay to its average when is too close to it we increment a counter. If the counter reaches a threshold we stop the simulation. The stop condition must be checked for all the nodes of the network.

The monitor Stop_steady_Network is a breakpoint monitor which checks if for each node we collect a satisfied count number (chosen 1000) of the msg_i_c.

4.1 Throughput

Figure 8 shows the throughput of the nodes versus the total requested load. We can see that the node 0's throughput is constant (100%) and is not affected at all values of the requested load. In fact, node 0 is a high priority node so its message will always be sent even if the throughput increases.

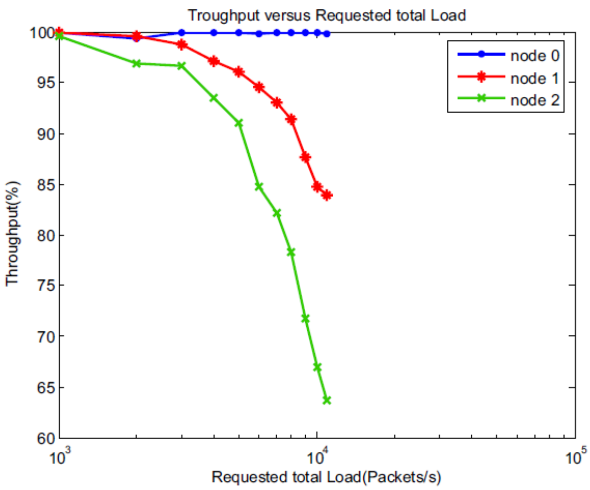

Figure 8: Throughput of all nodes versus the total load

However the others nodes can't achieve the requested throughput due to an increase of high priority messages. The impact of high priority message on the other nodes is clear on the Figure 8. The node 1's throughput is affected by the presence of higher messages priority from the load 2000 Packets/s. The influence is more important for the lowest priority messages (node 2). We note that at 10000 packet/s, the throughput of node 1 and 2 are affected and achieve only 84.6 and 66.9 % respectively.

4.2 Latency Delay

The Figure 9 shows that the latency (for all the nodes) starts to be constant, 93 μ s, from 10 packets/s till 100 packets/s. 93 μ s is the minimum time delay needed to transfer a packet over the network. From the load 100 packets/s, the latency delay increases for all nodes with the same manner. At 6000 packets/s the latency of node 2 considerably increases comparing to the others one.
We can also notice that the node 0's and node 1's latency are approximately close to each other.

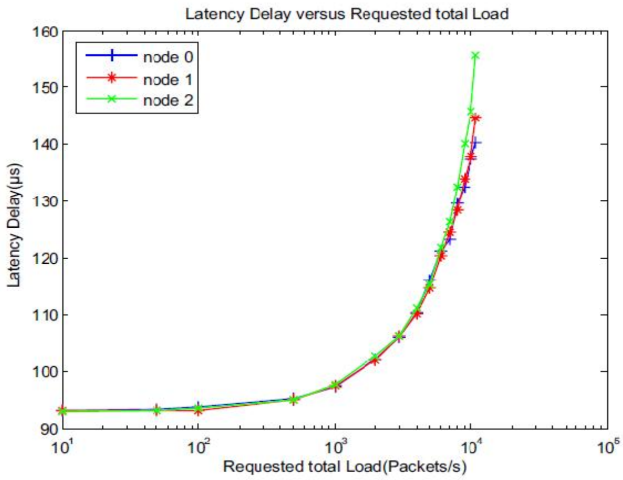

Fig. 9: Latency delay for all nodes versus the total load.

4.3 Loss probability

Figure 10 presents the global loss probability on the proposed architecture and the different message classes. We can see that the global loss probability increases even lowest priority node on the network so loss is easily feasible with the presence of higher priority messages. However node 1's loss probability, medium priority, is affected from 2000 packets/s and reaches a percentage loss of 4.4 % at 10000 packets/s. Note that at this load value, the percentage loss of node 2 is 16.5%. In addition at the maximum load, 10753 Packet/s, supported by the SCPN model the global loss is approximately 23%.

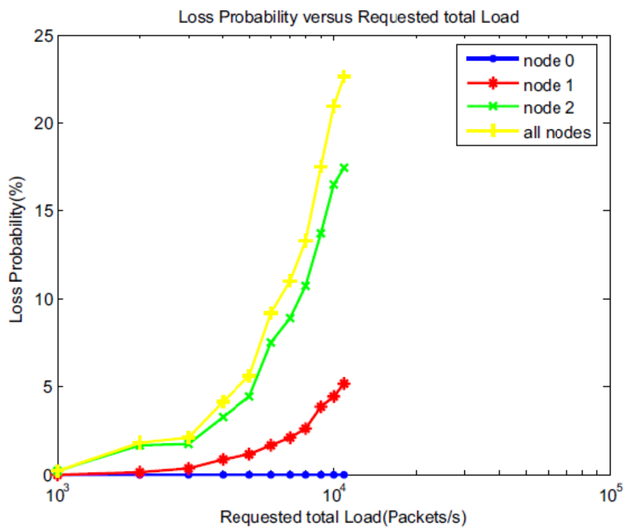

Fig. 10: Loss probability of all nodes versus the total load

5. Conclusion

In this paper, a VHDL implementation of a CAN Controller is realized. We also proposed the SCPN models of the overall switched fabric CAN network, evaluated by CPNTools.

The SCPN model of our new architecture is presented with three nodes at transmitter side. The model focuses on access mechanism scheme which is a key factor for this Switched fabric CAN Network. Three message priority classes were treated with a clear representation on ID field. Throughput, latency and loss probability were analyzed which reflects the real time aspect of our proposed architecture. Numerical values were given to choose the right value of load in accordance to the message priority for a correct system implementation.

The ACK field, the stuffing bits, and CRC field are omitted just for the simplicity presentation of the CAN arbitration access mechanism. These parameters can slightly modify the numerical values presented in this paper.

References

[1] Salem Hasnaoui, Oussema Kallel "A proof-of-concept Implementation of Modified CAN Protocol on CAN Fieldbus Controller Component"; Accepted oct. 2004, Revised Jan. 2005; AMI. Journal, Ref 03/08.

[2] G. Blair, G. Coulson, P. Robin, M. Papathomas, “An Architecture for next generation middleware”, Proc, 4th Annu, IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing, London, England,1998.

[3] Marko Bago, Siniša Marijan, Nedjeljko Perić; Modeling Controller Area Network Communication, proceedings of the Ninth Workshop on Practical Use of Coloured Petri Nets and the CPN Tools, October 20-22, 2008

[4] CUI Lian-cheng, ZHAO Zheng-fang, XU Xiao-ju2, WU Fang-ming, SHAN Wei-zhen “Real Time Performance Analysis of CAN Bus Based on TimeNET” The 3rd International Conference on Innovative Computing Information and Control (ICICIC'08), 2008.

[5] Prodanov, W.; Valle, M.; Buzas, R.; Pierscinski, H.”A Mixed-Mode behavioral model of a Controller-Area-Network bus transceiver: a case study” Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International Volume, Issue , 20-21 Sept. 2007 Page(s):67–72

[6] Heller, C. Reichel, R. “Enabling FlexRay for avionic data buses” Digital Avionics Systems Conference DASC '09,Oct 2009.

[7] FlexRay Communication System. [Online]. available: http://www.flexray.com

[8] Brett Murphy, Emmanuel Eriksson. “Fabrics and Publish-Subscribe Schemes: A Net-Centric Blend” COTS Journal Oct. 2009 <http://www.cotsjournalonline.com/articles/print_article/100148>

[9] Arshad, Nauman, Stewart Dewar, and Ian Stalker. “Serial Switched Fabrics Enable New Military System Architectures.” COTS Journal Dec. 2005 <www.cotsjournalonline.com/home/article.php?id=100438>

[10] Vedran Kordic “Petri Net Theory and Applications”; (ITECH Education and Publishing www.i-techonline.com. Chap3 pp51-82; February 2008)

[11] K. Jensen , “Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts” (Monographs in Theoretical Computer Science, Springer-Verlag, 1997).

[12] Dr. Rajive Joshi,” Using Switched Fabrics and Data Distribution Service to Develop High Performance Distributed Data-Critical Systems” The Journal of Defense Software Engineering. *April 2007.*

[13] John D. Pape, “Implementation of an On-chip Interconnect Using the i-SLIP Scheduling Algorithm”. (The University of Texas at Austin Chap 3 pp 11-15: December 2006)

[14] Oussama KALLEL, Sofiene DRIDI, Salem Hasnaoui “Modeling and Evaluating a CAN Controller Components Using Stochastic and Colored Petri Nets” International Review on Computers and Software. January 2009.

[15] Salem Hasnaoui, M. Hamdi, S. Tahar, “Throughput Performance Analysis of Wireless Control Area Networks and its Coupling to the Latency Time”, accepted for publication as a regular paper in Wireless Communications & Mobile Computing (WCMC) John Wiley Interscience Journal, April 2006, Revised September 2006, Ref WCMC-1105-353.To appear in 2009 (Editor's letter).

Mohamed Mazouzi was born in 1980 in Sfax, Tunisia. He received the degree in computer science engineering from The National School of Computer Sciences, Tunisia, in 2004, and the postgraduate research degree in New Technologies of Dedicated Computer Systems, in 2006, from National School of Engineering of Sfax. Since 2006, he is a PHD student in the Computer and Embedded Systems laboratory of National School of Engineering of Sfax. Since 2006, he occupied the post of a Technologist at the Higher Institute of Technological Studies of Gabes in Telecommunication Department. His research interests are industrial networks for real-time and distributed applications.

Oussama Kallel was born in 1977 in Sfax, Tunisia. He received the degree in electrical engineering from National School of Engineering of Tunis, Tunisia, in 2000, and the postgraduate research degree in communication networks, in 2002, from National School of Engineering of Tunis. Since 2002 he is a PHD student in the Computer and Communication Technologies Department of National School of Engineering of Tunis (El Manar University of TUNISIA). Since 2003, he occupied the post of an assistant at the Science Faculty of Bizerte in Physic department (Electronics Unit). His research interests are industrial networks for real-time and distributed applications and their implementations on FPGA component.

Salem Hasnaoui is a professor in the Department of Computer and Communication Technologies at the National School of Engineering of Tunis. He received the Engineer diploma degree in electrical and computer engineering from National School of Engineering of Tunis. He obtained a M.Sc. and third cycle doctorate in electrical engineering, in 1988 and 1993 respectively. The later is extended to a PhD. degree in telecommunications with a specialization in networks and real-time systems, in 2000. Salem is author and co-author of more than 40 refereed publications, a patent and a book. His current research interests include real-time systems, sensor networks, QoS control & networking, adaptive distributed real-time middleware and protocols that provide performance-assured services in unpredictable environments. Prof. Hasnaoui is the responsible of the research group "Networking and Distributed computing" within the Communications Systems Laboratory at the National School of Engineering of Tunis. He served on many conference committees and journals reviewing processes and he is the designated inventor of the Patent "CAN Inter-Orb protocol-CIOP and a Transport Protocol for Data Distribution Service to be used over CAN, TTP and FlexRay protocols".

Mohamed Abid is currently professor at Sfax University in Tunisia. He holds a Diploma in electrical engineering in 1986 from the University of Sfax in Tunisia and received his PhD degree in computer engineering in 1989 at University of Toulouse in France. His current research interests include Hardware-Software System on Chip co-design, reconfigurable FPGA, real time system and embedded system. Dr. Abid has authored/co-authored over 150 papers in international journals and conferences. He served on several technical program committees and as a co-organizer of several international conferences. He also served Guest Editor of special issues of two international journals.