International Journal of Computer Science, Communication & Information Technology (CSCIT)
Copyright – IPCO-2014
Vol.1, pp.7-13, 2014

AFDX-CAN Architecture for Avionics Applications

Nejla REJEB#1, Imene MHADHBI#2, Ahmed Karim BEN SALEM#3, Slim BEN SAOUD#4

#*LSA Laboratory, INSAT-EPT, University of Carthage, TUNISIA*
1rejebnejla2@gmail.com
2imene.mhadhbi@gmail.com
3ahmed.bensalem@ept.rnu.tn
4sbsaoud@yahoo.fr

Abstract— **Embedded avionics systems have recently emerged some limitations to respond to the rapid increase in the functionality requirements of new generation aircraft.**

In order to cope with these limitations, new avionics architectures have been designed to improve the efficiency in the whole avionic system. The current communication architecture of new generation aircrafts is a heterogeneous architecture based on the high rate backbone network AFDX (Avionics Full Duplex Switched Ethernet) interconnected to low rate data peripheral buses.

This paper aims to analyze a mixed AFDX/CAN architecture. We study how communication takes place in such a heterogeneous embedded network system seeking guaranteed performance. The goal with this case study has been to investigate necessary functions in the gateway nodes to support guaranteed end-to-end real-time communication. The global architecture must respect real-time requirements. Therefore, an analysis of the communication latencies per device has been made, in order to evaluate the real-time performance of the global network.

Keywords— **heterogeneous embedded networks, AFDX, ARINC 825, CAN, gateway, end-to-end delay**

I. INTRODUCTION

The progress of the avionics embedded architectures implies a significant increase in the complexity of electronic controls, and in the number of actuators and sensors in the current aircraft. All these innovations involve a great increase in the data flow between various and heterogeneous systems and thus, in the number of connections between functions.

To control this complexity, new avionics architectures, called Integrated Modular Avionics (IMA), as described in the ARINC 653 standard [1], have been designed to improve the efficiency in the whole avionic system and to answer the new requirements (centralization, determinism, higher rate, realtime, etc.).

In fact, nowadays aircraft have a completely new architecture that integrates different fields, applications, and heterogeneous networks. The last consists of different sub-heterogeneous networks (field busses, traditional avionics protocol such as ARINC 429 [2], sensor networks, open world network, etc.). While, the federator avionic technology remains the Avionics Full DupleX switched ethernet (AFDX) [3]-[5] which represents a redundant and reliable Ethernet network, developed and standardized by the European industrial avionics and in particular by Airbus.

The heterogeneity of such interconnection system involves different requirements to satisfy the Quality of Service (QoS) in terms of delay, jitter, bandwidth, message loss and integrity Therefore, it requires gateways to solve the problem of different avionics busses dissimilarity. As a consequence, design certification and network performance analysis require new study techniques.

In this paper, as heterogeneous network architectures, we consider the avionics field buses CAN [6] that is already integrated into the aircraft interconnected to the AFDX. We have studied different operations for the gateway nodes including fragmentation and addressing. Further on, we have studied how to incorporate real-time analysis to be able to guarantee end-to-end delay performance. It is mandatory to analyze the end-to-end delays over a heterogeneous path. It includes the timing analysis of the bridging strategy between the different technologies. Whereas the data flow is transmitted by more than one technology, it is necessary to analyze the possible end to end delay for such a heterogeneous network. The gateway impact on the real time behavior is a major challenge in the design process of heterogeneous embedded system. Therefore, we have been to investigate necessary functions in the gateway nodes between different networks and to propose a solution to support guaranteed end-to-end real-time communication.

The next section presents briefly AFDX and CAN technologies. The third section describes performance evaluation techniques. Then, in the fourth section, we consider CAN/AFDX architecture and propose CAN/AFDX bridging strategies. Then, the end-to-end delay is analyzed. Section 5 concludes the paper and presents some ideas for future works.

II. COMMUNICATION TECHNOLOGIES

We present the tow communication technologies we intend to use, CAN and AFDX.

A. *AFDX*

AFDX technology [3]-[5] brings a number of improvements such as higher data speed transfer and much less wiring, thus improve determinism and guarantee bandwidth.

AFDX is a standard that defines the electrical and protocol specifications (IEEE 802.3 and ARINC 664, Part 7) for exchanging data between avionics subsystems. It is used as the main avionics data bus network. Based on commercial 100 Mbit/s switched Ethernet, AFDX uses a special protocol for deterministic timing and redundancy management to provide secure and reliable communications of critical and non-critical data.

when an application sends a message from the source sub-system to the destination application, the source end system, AFDX switch and end system destination are configured to deliver the message to the appropriate ports.

The inputs and outputs of the networks are called End Systems (ES) which are interconnected by switches. Each end system is connected to exactly one port of an AFDX switch and each port of an AFDX switch can be connected at most to one end system. All the end systems and switches support First-In-First-Out (FIFO) queuing. All the links in the network are full-duplex.

1) *Virtual Link*

Virtual Links (VL) [4], [5] standardized by ARINC-664 are the central feature of an AFDX network. A VL is a virtual logic connection with a unicast source and multicast destination.

For the purpose of determinism, virtual links specify a static path for each data flow. Data is transmitted according a Virtual Link Identifier VLID.

A VL is characterized by two parameters to describe the performance:

- Bandwidth Allocation Gap (BAG): is the primary bandwidth control mechanism. The minimum time interval between consecutive frames of the corresponding VL (fig 3), is a power of 2 value in the rank [1,128],
- Minimum and Maximum Frame Length (Smin and Smax): the Ethernet frame length adopted by AFDX is between 64Kb to 1518Kb.

2) *Sub-Virtual Links*

A virtual link can be composed of a number of Sub-Virtual Links Each Sub-VL [3], [4] has:

- a dedicated FIFO queue,
- a round robin algorithm working over IP fragmented packets.

3) *AFDX End System*

The end system ES is the AFDX element which provides an "interface" between the subsystems and avionics AFDX interconnection (fig 1).

An ES receive messages in it communication ports from avionics devices, encapsulating them within UDP, IP, and Ethernet headers and placing them on their adequate Virtual Link queue.

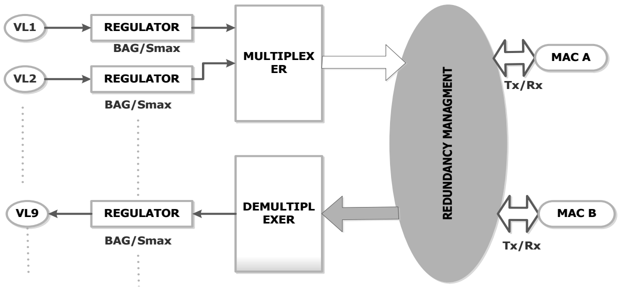

Fig 1: End system model

4) *AFDX Switch*

The switch is the most important equipment in AFDX network defined by the standard 802.1D [7]. Each switch has to filter, police, and mainly forward the arriving packets their destination addresses throw its appropriate ports as shown in Fig 2. The switch examines a forwarding table to determine the corresponding Tx port for every Rx packet according to the correspondent VLID.

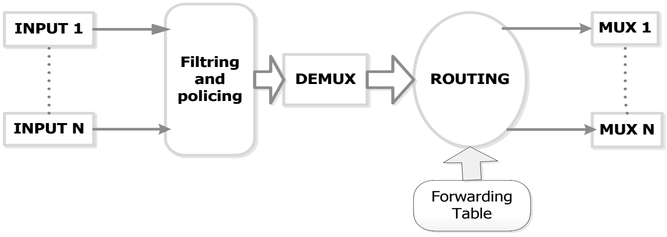

Fig 2: AFDX Switch model

5) *Frame Format*

The AFDX frame format is described in Table 1. The destination and source addresses contain the MAC addresses for the ES. The MAC destination address carries the VLID in the last 16 bits. IP address information is contained in the IP Structure block. The UDP structure identifies the appropriate application port. The AFDX payload ranges from 17 to 1471 bytes.

TABLE I: Frame format

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	20 bytes	8 bytes	17 to 1471 bytes	1 byte	4 bytes	12 bytes
Preamble	Start Frame Delimiter	Destination Address	Source Address	Type IPv4	IP Structure	UDP Structure	AFDX Payload	Seq Number	Frame Check Seq	Interframe Gap

6) *Maximum Jitter*

The jitter is defined as the difference between the beginning of the BAG and the first bit of the frame being sent (fig 3).

To guarantee determinism, the maximum allowed jitter on each VL at the output of the end system should respect the two following formulas*:*

$$\text{Max.Jitter} \le 40\mu s + \frac{\sum_{i\in\{setofVLS\}} \left(20bytes + L_{MAX}\right)\times 8}{N_{BW}}$$

$$\text{MAX.Jitter} \leq 500\mu s$$

Where:

- N_{BW} is the speed of the Ethernet link in bits/s
- 40 μs is the typical minimum fixed technological jitter
- 500 μs is the total jitter that is allowed to exceed

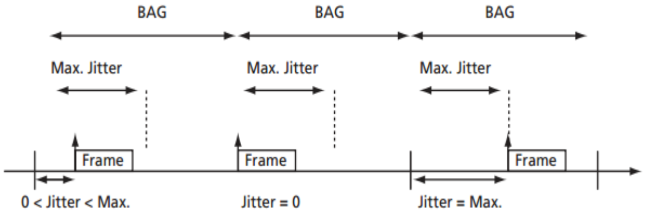

Fig 3: BAG and Jitter

B. CAN

CAN is a serial communication protocol suited for networking sensors, actuators and other nodes in real-time systems. The CAN specification [8] defines several versions of the protocol for the physical and the data link layer. For shorter bus lengths, the maximum data rate of 1Mbit/s can be used. The CAN addressing system is based on message identifiers: a frame does not have a destination nor a source address. Frames are broadcasted on the bus. Stations get the frames they are interested in by a filtering process of the identifiers. CAN is a broadcast bus using an object oriented approach for data transmission.

The frame format [8] is depicted in Table 2.

TABLE 2: CAN frame (sizes in bits)

1	**11/29**	**1**	**1**	**1**	**4**	**0 to 64**	**16**	**2**	**7**	**3**
SOF	***Identifier***	***RTR***	***IDE***	***r0***	***DLC***	***Data***	***CRC***	***ACK***	***EOF***	***IFS***

The most important fields are the following:

- the identifier field, which as mentioned earlier identifies the data contained in the frame so the data payload can be processed correctly in the receiving nodes. CAN supports two versions of identifiers with different lengths (11 bit and 29 bit), referred to as "standard" and "extended" identifiers,
- the DLC field which gives the length (in bytes) of the data field,
- the data field :which is the payload of the frame (between zero and eight bytes).

CAN uses a sophisticated error detection and handling protocol, consisting of a 15-bit Cyclic Redundancy Check (CRC), frame structure, data acknowledge checking and bus signal monitoring. Any node on the network which detects an error during data transmission or reception immediately sends an error flag. This error flag destroys the current (faulty) message and causes the transmitting station to abort the transmission. All nodes then disregard the current message and check to see if they were the cause of the error.

The collisions on the bus are resolved following a CSMA/CR protocol (Carrier Sense Multiple Access / Collision Resolution) thanks to the bit arbitration method. When two or more stations start a transmission simultaneously, the one with the highest priority identifier (lowest value) wins and the others stop their transmission. This is implemented by collision detection on a bit by bit basis. When a station transmits 1 (recessive bit) and detects 0 (dominant bit), it knows that a frame with a higher priority is being transmitted and, consequently, it immediately stops transmission. This mechanism guaranties strict priority order on identifiers, provided identifiers are unique. It implies limitations of the bandwidth and the maximal length of the bus.

CAN increasingly found its way into aerospace applications because of its cost effective and efficient networking capability for LRUs (line replaceable units) [9]. The ability of CAN to transmit data, across a shared shielded twisted pair cable, has advantages in terms of weight savings at the aircraft integration level. Additionally, the CAN physical layer protocol specification provides error recovery and protection mechanisms making it attractive to aviation applications for all sorts of functions including flight deck systems, engine control and flight control systems.

While CAN components and technology have served the automotive industry well over the years, there are certain aspects that need to be adapted to the airborne environment. Specifically the CAN protocol requires definitions to control the priority and separation of message delivery across the network suitable to meet the needs of aerospace applications. At the airplane level there is a need to standardize aspects of the protocol at the system level to ensure interoperability across system and network domains. These needs were met first by the CANaerospace standard [9], which was established in 1998 and is widely used within the general aviation world, then by the ARINC 825 standard [10] which was defined by the CAN Technical Working Group of the Airlines Electronic Engineering Committee.

Nowadays, general aviation system architectures employ CAN as one of the major avionics networks. It is used to link sensors, actuators and other types of avionics devices that typically require low medium data transmission volumes during operation.

III. PERFORMANCE EVALUATION TECHNIQUES

Many methods for network performance evaluation have emerged to design, guarantee the quality of service and evaluate time performance for a given network. Network evaluation utilizes the actual network, an emulated network or a model of the network. These methods are classified into two main groups: performance modelling and performance measurements.

Different methods used for the study and analysis of a temporal network avionics are made for homogeneous networks in several researches. This study focuses on the area of real-time performance evaluation of heterogeneous avionics network.

A. Performance Modelling

Performance modelling is typically used when actual systems are not available for measurement or if the actual systems do not have test points to measure every detail of interest. Researchers utilize knowledge about the interactions of network components to understand and explain the workings of a computer network via a conceptual model. Models are partitioned into simulation models or analytic models. . Both modelling techniques tend to rely on queuing theory. They rely on simplifying assumptions that enable the model to capture important characteristics of networks.

1) Simulation Method

Simulation attempts to reproduce the behavior of the network in the time domain. This approach needs a realistic model based on queuing theory. Simulation is essentially a numeric solution that utilizes systems of equations and data structures to capture the behavior of the simulated network in terms of logical conditions. This approach allows taking into account configurations of industrial network size. However, for simulation, we must gain sufficient confidence in the sense that all the scenarios retained after the method application are representative and provide a valid distribution delays throughout.

Several discrete events network simulators were used in the literature for simulation of a homogeneous AFDX network (eg. NS2, NS3, Opnet, QNAP2) [11]- [14] and CAN bus [15].

2) Analytical Method

Similar to simulation models, analytic models involve systems of equations. Analytic models of computer networks usually start with a network of queues model and develop a system of equations that may yield a closed form solution. They are based on mathematical models to extract performance criteria. Among these methods, there are deterministic and probabilistic techniques. The first techniques compute conservative bounds for parameters they evaluate; while the second techniques provide all possible parameters values matching probabilities of achieving them.

Two methods are used for the deterministic bounds computing: the network calculus [16]-[18] and the trajectories method [19], [20]. These approaches are a pessimistic analysis, since it is based on pessimistic assumptions. Indeed, all these methodologies have complementary probabilistic extensions [21]: a probabilistic upper bound has been calculated for the crossing time. These extensions are based on the same assumptions used on the network calculus and the trajectories methods. Thus, they don't solve the problem of pessimism results.

B. performance measurement

Performance measurement is possible only if the system of interest is available for measurement and only if one has access to the parameters of interest.

The overall objective of the computer network measurement study guides the choice of performance indices to be measured. Metrics are either direct or indirect indices. Indirect indices require some type of data reduction process to determine metric values.

Due to the large data volume associated with network traffic, measurement of computer networks often involves filtering of data or events (e.g., It is common for network measurement tools to only retain packet headers for off-line analysis).

When the measurement strategy involves probabilistic sampling, the duration of the experiments is determined using confidence interval techniques. While hardware probes provide the best quality measurements, they are expensive and not always available.

IV. CASE STUDY: HETEROGENEOUS CAN / AFDX

A. CAN / AFDX architecture

We consider the AFDX network as a backbone network to interconnect the critical avionics systems, and to dissociate the sensors and actuators from their attached end systems. As described in fig 5, the peripheral sub networks CAN are specific either for sensors or for actuators in order to avoid the possible contentions between the flows coming from AFDX to CAN and from CAN to AFDX that can lead to performance degradation. The obtained clusters are interconnected via specific gateways, called Remote Data Concentrators (RDCs) and standardized as ARINC655 [22]. RDCs are distributed throughout the aircraft and function as a gateway between the aircraft sensors and actuators, and the avionics processing resources.

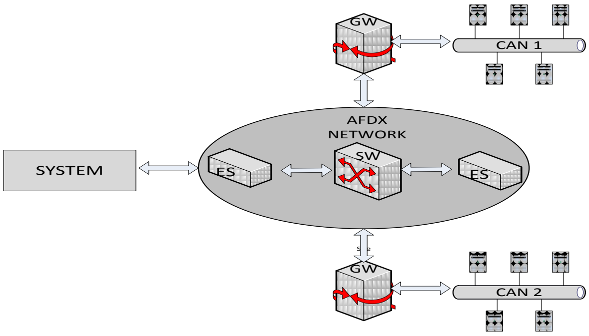

Fig 4: Heterogeneous AFDX- CAN

Our case study, as illustrated by Fig 4, is consisted of the following sub-systems:

- AFDX network: AFDX ES interconnected by AFDX SW,
- GW: RDC that allows the communication between the avionics world and the peripheral network (sensor network, open world, etc.),
- tow CAN busses: CAN 1 is used for data acquisition from sensors and CAN 2 for data transmission to the actuators, that typically require low medium data transmission volumes during operation.

1) *Traffic over the network*

All the traffic is transmitted on CAN data busses or/and AFDX network. Three kinds of traffic have to be considered:

- local CAN traffic: frames local to a CAN bus: they only have to be transmitted over this bus. They are produced by a station on a CAN bus and consumed by stations all on the same CAN bus,
- local AFDX traffic: frames local to the AFDX network: they only have to be transmitted over the network from a ES source to a ES destination through the switch,
- global traffic: frames from a local station of a CAN bus to a calculator of the AFDX network and vice versa: they have to be transmitted by the CAN station , received by the bridge associated with the CAN bus, transmitted over AFDX, received by the calculator.

2) Bridging Strategies for global traffic

For the global traffic, we need bridging equipments (GW) to handle the communication between the two protocols CAN and AFDX which are very different:

- The available bandwidth: 1MBs or less for CAN, 100 MBs for AFDX,
- The addressing system: identifiers associated to data for CAN, MAC Addresses of station (VLID) for AFDX, therefore, we need a global memory addressing scheme which will have the address of every node in the heterogeneous embedded network.
- Different MTU (Maximum Transfer Unit) packet size, therefore, there is a need for fragmentation and reassembly function. The data encapsulated in a frame: between 0 and 8 bytes for CAN, between 64 and 1518 bytes for AFDX,
- The collision resolution: ADFX is a deterministic network, CSMA/CR collision protocol for the CAN bus.

The gateway nodes are designed for heterogeneous embedded protocols which will allow these networks to communicate with each other with the help of different translation functions. The GW has principally to perform protocol conversion which includes extracting the payloads of the incoming messages and then adding the correct protocol headers before sending them to their destination network. Heterogeneous protocols require to operate on Layer 7 to connect two different networks. Gateway makes communication possible between two different architectures and protocols. They encapsulate and convert the data from one network to another. In the gateway, all layers of OSI model are included (1 to 7). The protocol structure of the proposed gateway is shown in fig 5. The conversion function is handled on the application level. The CAN/AFDX gateway include a CAN-UDP Protocol converter, which converts CAN messages to UDP packets, and vice versa, as well as routing CAN messages.

The gateway node needs to reformat the packet to meet the requirements of destination network. In order to reformat the packet the following gateway operations will be performed in our heterogeneous embedded network architecture: Data formatting and Addressing.

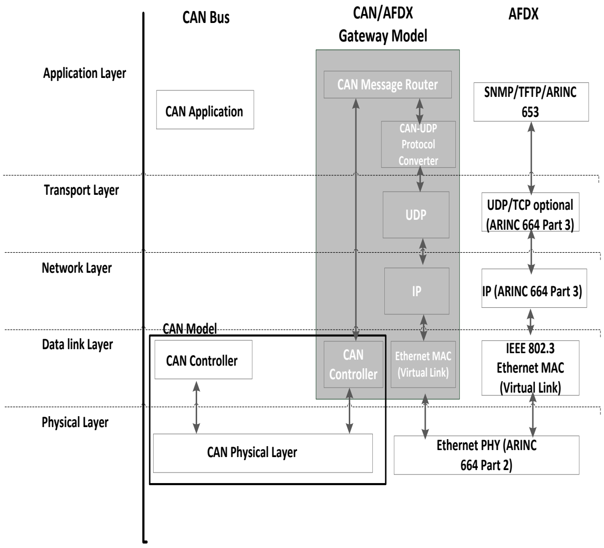

Fig 5: Gateway Protocol structure

The gateway node is based on a conversion function that convert each CAN frame to an AFDX frame and vice versa. Each frame received on the gateway input interface is decapsulated to extract the payload. Then, a required header is identified and added to the extract payload according to a static routing table. This function is simple to implement, but can induce a big latency in both input and output interfaces. The gateway node (RDC) becomes one of the main node that needs to be reconfigured to improve the real time performance of the global system.

3) *Case Study end-to end Evaluation Analysis*

When communication across gateway nodes takes place in a heterogeneous embedded networks system, the investigation of the end-to-end delay from start to end becomes necessary to guarantee performance.

Thus, the approaches described on section III, which have already been made to the homogeneous network AFDX for the communication latencies analysis, should be extended and generalized to a global heterogeneous network.

The study of a communication medium determinism, in particular temporal determinism, requires the end-to-end latency evaluation: the delay between the message input in the communication stack of the transmitter module (Network 1) and the outlet in the communication stack of the receiver module (Network 2). The determination of an upper bound of the end-to-end latency is a major constraint in the certification process.

If it appears that the estimation of end-to-end latency through IMA must be comprehensive, this assessment, however, faces problems of complexity induced precisely by global character.

So, the study of such a heterogeneous network and the analysis of the gateways characteristics and their impact on the performance of end-to-end delay becomes a major challenge in the design process of heterogeneous embedded systems. However, the few studies focusing on avionics heterogeneous networks have ignored the impact of gateways on the system performance [23]-[26]. Therefore, we have chosen to focus on the study of heterogeneous network, taking into account the impact of the interconnection equipments on end-to-end time system performance.

1) Gateway impact on the end- to-end delay

A gateway approach for achieving semantic interoperability becomes complex and may require long processing times. These delays are equal to the payload extraction and mapping latency.

The gateway mapping strategy according to their functions affects the duration of the message latency at the gateway. So, this duration cannot be considered constant, and the determination of such a delay is necessary for the end to end delay evaluation of a global system.

Gateway uses the most common queuing algorithm FIFO. The latency on the gateway may be defined as:

$$D_{GW} = D_{Rx} + D_{O.GW} + D_{Tx}$$

Where:

- D_{Rx} is the delay that an incoming message has to wait until the message is served from the input buffer
- $D_{O.GW}$ is the gateway operating time
- D_{Tx} is the delay until an outgoing message on the output buffer can be sends in the destination domain

2) AFDX end to end delay analysis

Fig 8 illustrates an AFDX configuration.

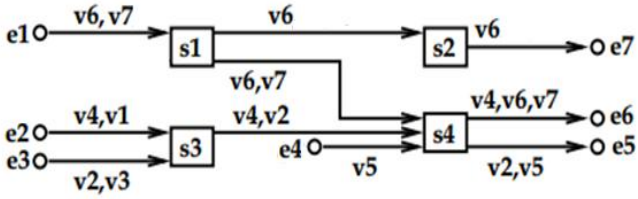

Fig 8: An illustrative AFDX configuration

The AFDX end-to-end delay may be determinated as following:

$$D_{AFDX} = D_{ES} + \left(nb_l \times D_l\right) + \left(nb_{SW} \times t_{SW}\right) + \sum_{SW \in \{setofswitches\}} D_{SW}$$

Where:

- D_{ES} is the delay in the source end system output buffer, nb_l is number of links on a VL path
- D_l is the transmission delay over a link
- nb_{SW} is number of switch on a VL path
- t_{SW} is the delay in a switch from an input port to an output port is considered as a constant = 16µs
- D_{SW} is the delay in SW output port buffer

3) Global end-to- end delay definition

Indeed, the end-to-end delay (D_{eed}) becomes:

$$D_{eed} = D_{AFDX} + D_{GW} + D_{ARINC825}$$

Where:

- D_{AFDX} is the end to end delay for a given AFDX message crossing the AFDX network, which may be calculated using timing performance approaches described in section III,
- D_{GW} is the duration a frame might be delayed in the gateway,
- D_{CAN} is the propagation time across the CAN bus for a given message to be received by the gateway from a sensor or to be transmitted from the gateway to an actuator.

V. CONCLUSIONS

For avionics embedded applications, it is essential that the communication network fulfills certification requirements (real time performance, determinism, etc.)

In this paper, we analyse a heterogeneous avionics networks AFDX-CAN in order to define, as realistic as possible, a real-time performance evaluation. We propose a bridging strategy that allows the communication between these two heterogeneous embedded networks. Then, in order to deal with the worst case performance analysis of such network, a global end to end delay analysis has been done. In fact, the end-to-end delay must take into account the impact of the interconnection equipments. The use of gateways may increase the communication latencies and real time constraints have to be verified.

To evaluate the global network (AFDX-Gateway-CAN), we propose to opt for the simulation approach. This constitutes the objective of our running work.

Moreover, the optimization of an avionic gateway can be considered to improve the avionic network real-time performance.

REFERENCES

[1] *ARINC 653, An Avionics Standard for Safe, Partitioned Systems, 2008.*

[2] *ARINC 429, Protocol Tutorial*, 2004.

[3] *ARINC 664, Aircraft Data Network, Part 1: Systems Concepts and Overview*, 2002.

[4] *ARINC 664, Aircraft Data Network, Part 7: Deterministic Networks*, 2003.

[5] *AFDX/ARINC 664 Protocol Tutorial*, GE Fanuc Embedded Systems Information Centers, 2007.

[6] S. Schneele, C. Heller, M. Mawire, H. Thompson, C. Mital, C. Thiry, ”Use of automotive data bus in avionics: CAN and FlexRay”, in MOET Project Consortium, 2009.

[7] *IEEE 802.1D, Local and Metropolitan Area Network: Media Access Control Level Bridging*, 1998.

[8] R. B. GmbH, *CAN specification Version 2,0*, 1991.

[9] Stock, M.: *ARINC Specification 825 -The General Standardization of CAN for Airborne Use*, CAN Newsletter 12/2009.

[10] R. Knueppel, “Standardization of CAN networks for airborne use through ARINC 825“, iCC,2012.

[11] J.L Scharbarg, C. Fraboul. ”A generic simulation model for end-to-end delays evaluation on an avionics switched Ethernet”, in IFAC International Conference On Fieldbuses & Networks in Industrial and Embedded Systems (FeT 2007), Toulouse, 07/11/2007-09/11/2007,LAAS, p. 159-166, novembre 2007.

[12] A. Ben Achballah and S. Ben Saoud, “investigating the use of AFDX protocol as a network-on-chip”, in IEEE int. conf. DTIS, Gammarth, Tunisia, 2012.

[13] J.L. Scharbarg and C. Fraboul, “Simulation for end-to-end delays distribution on a switched Ethernet”, in Emerging Technologies & Factory Automation, 2007. ETFA. IEEE Conference on, pages 1092–1099, September 2007.

[14] L. Scharbarg and C. Fraboul, “Study on Real-Time Performance of AFDX Using OPNET”, in IEEE Control, Automation and system Engineering (CASE), 2011.

[15] J. Hao, J. Wu, C. Guo, “Modeling and simulation of CAN network based on OPNET”, in IEEE Communication Software and Networks (ICCSN), 2011.

[16] R. Cruz, ”A calculus for network delay”, part I. IEEE Transactions on Information Theory, 37(1):114.131, January 1991.

[17] J. Grieu, **“**Analyse et évaluation de techniques de commutation Ethernet pour l’interconnexion des systèmes avioniques**“,**. PhD thesis, Institut National Polytechnique de Toulouse, Toulouse, France, September 2004.

[18] F. Frances, C. Fraboul, , **“**Using network calculus to optimize the AFDX network**“, i**n European Congress on Embedded Real-Time Software (ERTS 2006), Toulouse France, 25/01/2006-27/01/2006, SIA/3AF/SEE, (support électronique), 2009.

[19] H. Bauer, J.L Scharbarg, C. Fraboul ”Applying trajectory approach to AFDX avionics network”,in Euromicro Conference on Real-Time Systems (session Work in Progress) (ECRTS09 2009), Dublin, 01/07/2009-03/07/2009, IEEE, p. 57-60, juillet 2009.

[20] H. Bauer, J.L Scharbarg, C. Fraboul, **“**Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach“, in IEEE Transactions on Industrial Informatics, IEEE, 2010 .

[21] J.L Scharbarg, F. Ridouard, C. Fraboul, ”A probabilistic analysis of end-to-end delays on an AFDX avionic network”**,** in IEEE Transactions on Industrial Informatics, IEEE, Vol. 5 N. 1, p. 38-49, février 2009.

[22] *Remote Data Concentrator (RDC) generic description, ARINC report 655,1999.*

[23] J. Ermont, J.L Scharbarg, C Fraboul, **“**Worst-case analysis of a mixed CAN/Switched Ethernet architecture“, in International Conference on Real-Time and Network Systems (RTNS 2006), Poitiers, 2006.

[24] J.-L. Scharbarg, M. Boyer, and C. Fraboul, “CAN-Ethernet architectures for real-time applications.”, in IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2005.

[25] H. Ayed, A. Mefdaoui, and C. Fraboul, “Gateway Optimization for an Heterogeneous avionics network AFDX-CAN.”, in IEEE Real-Time Systems Symposium (RTSS), 2011.

[26] P. Pop, P. Eles, Z. Peng, and T. Pop, “Analysis and optimization of distributed real-time embedded systems”, ACM Transactions on Design Automation of Electronic Systems, 2006.