AFDX simulation Based on TTEthernet Model under OMNeT++

Nejla REJEB
National Institute of Applied Sciences and Technology/ESPRIT
University of Carthage
Polytechnic School of Tunisia, LSA laboratory
Tunis, Tunisia
Nejla.rejeb@esprit.tn

Ahmed Karim BEN SALEM
Dept. of Electrical Engineering,
Higher Institute of Medical Technologies
University of Carthage
Polytechnic School of Tunisia, LSA laboratory
Tunis, Tunisia
ahmed.bensalem@ept.rnu.tn

Slim BEN SAOUD
University of Carthage
Polytechnic School of Tunisia, LSA laboratory
Tunis, Tunisia
Slim.bensaoud@gmail.com

Abstract—**The new generations of aircraft embark, growingly, avionics systems and functions to increase passengers' security and comfort. Therefore, a huge increase in data traffic is generated. Higher rates with more interconnections become necessary. However, Conventional avionics communications busses cannot meet these new needs. They are prompting avionics manufacturers (Airbus and Boeing) to install on their aircraft high speed communication networks based principally on switched Ethernet technology. Currently, the communication architecture of the latest aircraft generations is based on the AFDX (Avionics Full Duplex Switched Ethernet) backbone interconnected principally to data peripheral bus. The aim of our paper is to study and evaluate these new network architecture real-time performances. We are focused on the assessment based on simulation approach using OMNet++ simulator. We propose an illustrative example of the AFDX network that already exists on the literature. Firstly, the AFDX network end-to-end delay is determined by simulation. Then, we analyze and compare our results with other works. Thus, the AFDX performance is verified and validated through the simulation.**

Keywords— AFDX; TTEthernet; simulation; OMNeT++; performance evaluation; end-to-end delay.

I. INTRODUCTION

Avionics communication architecture becomes more and more complex due to the huge flow of exchanged data and the increasing number of interconnected subsystems. On the other hand, it becomes critical, for avionics applications, that the communication architecture fulfils the emerging necessities concerning modularity and performance (delay, jitter, bandwidth, message loss and integrity). Therefore, new aircrafts generations have a fully new architecture based on a high rate backbone network AFDX (Avionics Full Duplex Switched Ethernet) [1]. In fact, Ethernet has become the unavoidable trend in the new avionic generation electronics system thanks to its huge achievements in the industry. Based on the commercial Ethernet, full-Duplex Switched Ethernet has been enhanced to fill the real-time and deterministic requirements of the avionic field [2].

So, the new AFDX architecture, responding to the new generation of aviation electronics system demands has been integrated on the modern large aircraft such as A380 and B787. It has been standardized by the aircraft manufacturer under ARINC 664-part7 [1] thanks to its high transmission rate and high reliability. Indeed, principal avionic network requirements are a deterministic real-time network with high data transmission rate. Thus, the performance evaluation of such AFDX network is very important. To do this, three methods have been used on literature: field test, analytical methods and network simulation [2, 3, 4, 5]. However, the simulation method can give the performance parameters for comparison and analysis and allows much more comprehensive network characteristic. Therefore, we have considered this simulation approach to evaluate the avionic AFDX network.

This paper is organized as follows: section 2 presents the AFDX network characteristics. Then, in the third section a description of the AFDX model based on the TTEthernet is given. First, a comparison between different simulators is provided. Then, the TTEthernet model developed under OMNet++ is studied. On section 4, a test and simulation of an illustrative example of AFDX network under the OMNeT++ simulator and the end to end delay evaluation is presented. Section 5 concludes the paper and presents some ideas for future works.

978-1-5090-6634-6/17/$31.00 ©2017 IEEE

II. AFDX AVIONIC NETWORK CHARACTERISTICS

AFDX technology [1] brings a number of enhancements, such as higher data transmission rate and much less wiring, thus improve determinism and guarantee bandwidth. AFDX is a standard (IEEE 802.3 and ARINC 664, Part 7) that determines the electrical and protocol specifications for data exchange between avionics subsystems; based on commercial 100 Mbit/s switched Ethernet. It represents the main avionics data bus network used to procure secure, deterministic and reliable communications of critical and non-critical data.

A. AFDX switch

The switch is the most important equipment in AFDX network defined by the standard 802.1D. Its operation is based on the "Store and Forward" mode. Indeed, once the frames are received on the input links, the switch begins by analyzing them (filter and police) and then forwards them to the corresponding output (s) port (s).

If several frames occur at the same time to the same output port, those which are not yet transmitted is stored in a FIFO (First In First Out) queue. The switch examines a forwarding table to determine the corresponding Tx port for every Rx packet according to the correspondent VLID. *Fig. 1* illustrates the switch structure.

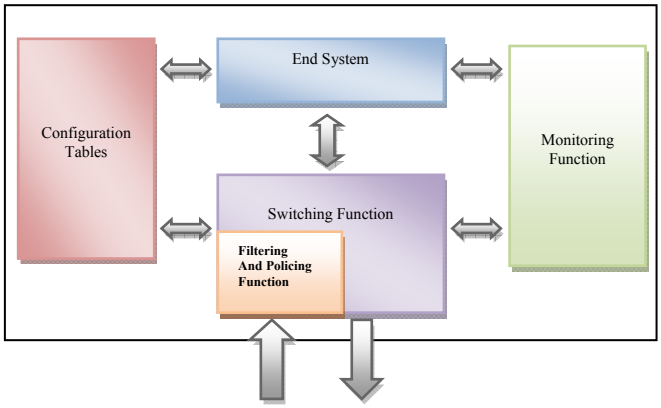

Fig. 1. AFDX switch

B. AFDX End System

The end system ES is the AFDX element which provides an "interface" between the subsystems and avionics AFDX interconnection (*Fig. 2*). Indeed, an ES receives messages in it communication ports from avionics devices. First, it encapsulates them within UDP, IP, and Ethernet headers. Then, it puts them on their adequate Virtual Link queue.

Several operations are performed by ES (structuring data to send it in frames, fragmentation/defragmentation frames, filter frames, or duplication of the frame on the redundant network, etc.) before entering or leaving a frame on the AFDX network. Its communication ports are standardized by the ARINC 653 [7] standard. These ports have two types:

- The Sampling Port: This port has a queue of one place. When an application sends a new frame to the port, the old frame is eliminated.
- The Port Queuing: This type has a waiting queue of several places. The new incoming frames are stored in queue. The first arrival is the first to be transmitted over the network using the FIFO service policy.

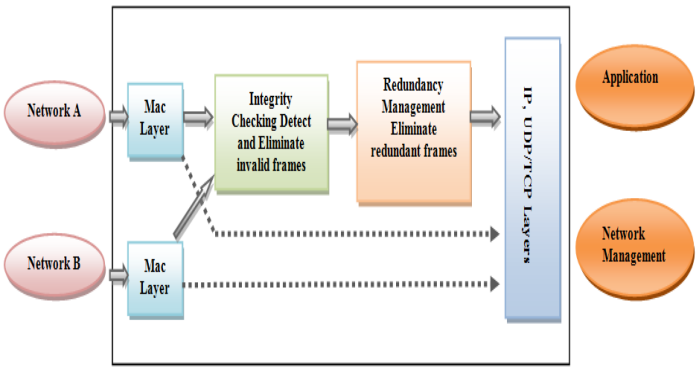

Fig. 2. AFDX End System [1]

C. Virtual Link

Virtual Links (VL) [1] standardized by ARINC-664 are the central feature of an AFDX network. The concept of AFDX Virtual Link (VL) offers the possibility to assure, for each traffic flow, a guaranteed bandwidth [1]. The VL is a virtual multicast channel that originates from a single source such as ES and delivers its packages for a fixed set of ES, as shown in *Fig. 3*. To guarantee deterministic transmission, virtual links define a static path for each data flow. A Virtual Link Identifier VLID is specified for each transmitted message.

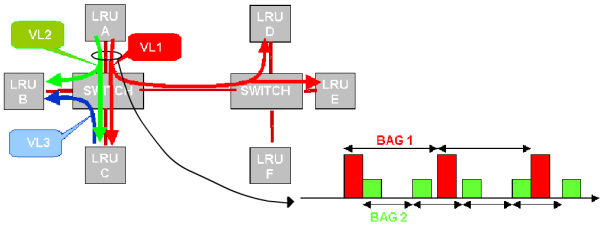

Fig. 3. AFDX Virtual Link [1]

Each VL is characterized by [1]:

- A single source ES,
- A static route to one or more ES destination,
- A unique identifier (VLID)
- A BAG (Bandwidth Allocation Gap), ranging from 1 to 128 milliseconds, representing the minimum delay between the emission of two consecutive frames on the I/O source,
- A frame with minimum size S_{min} and max size denoted S_{max}, between 64 and 1518 bytes which is the

size of the largest frame which can be sent during each BAG. If the frame of a VL generated by the ES does not respect the BAG defined, then it is deleted. *Fig.4* depicts this situation.

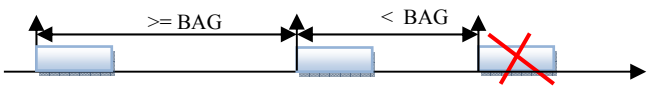

Fig. 4. BAG Illustration

So, a VL has a maximum flow equal to the maximum size of its frames, divided by its BAG (Smax /BAG).

D. *AFDX frame*

The AFDX frame [1], shown in *Fig. 5*, is based on the Ethernet frame.

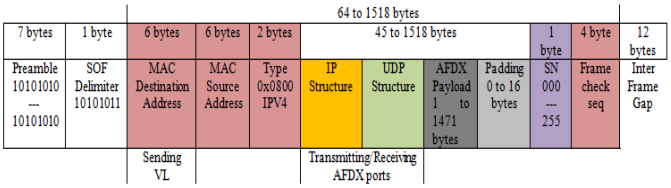

Fig. 5. AFDX frame format [1]

III. AFDX model based on TTEthernet

A simulation model of AFDX network can be very advantageous. Indeed, the real time performance analysis of such a network becomes necessary since AFDX network must fit real time requirements of avionics systems. Also, it is mandatory to better understand the real behavior of AFDX network. For this purpose, several research works have been conducted using different simulation platforms. For examples, one model was developed on the QNAP2 network simulator [8], a second was developed on the NS2 simulator [9]. Another one is based on the network simulator OPNET [10]. Recently, a model based on the TTEthernet (Time- Triggered Ethernet) was built on OMNeT++ [11, 12].

A. *Simulators comparaison*

The last simulators used in the different research activities represent various characteristics in terms of performance and scalability. Each one has advantages and disadvantages and may be appropriate for certain circumstances. Table 1 compares synthetically the properties of these different simulators networks studied. The choice of the simulator is therefore a delicate question, and depends largely on usage and requirements.

However, if the scalability and extensibility are the main concerns, OMNeT++ remains an interesting choice because it represents the only open source simulator of use and extension and it allows handling and modeling complex case study. Moreover, we have especially opted to use it because it already incorporates switched Ethernet and TTEthernet models used in our case study.

TABLE I. SIMULATORS COMPARISON

simulator / parameter	OPNet	NS-2	NS-3	OMNeT++
Langage	C/C++	C++/OTCL	C++/python	C++
License	commercial	Open source	Open source	Open source
Available Module	Wired, Wireless, Adhoc and WSN.	Wired,Wireless, AdHoc, and Wireless Sensor Networks **(-)Ethernet not available**	Wired ,Wireless, Adhoc and Wireless Sensor Networks **(-) Ethernet not available**	Wired, Wireless, Adhoc and Wireless Sensor Networks **(+)Ethernet/TTEthernet available**
Graphic interface	good	limited	good	Very good
Parallelism	yes	no	no	yes
Scalability	medium	limited	limited	large
Extensibility	Very good	good	good	excellent
Complexity	complicated	Very complicated	Very complicated	simple
Documentation	good	excellent	excellent	excellent

B. *TTEthernet Model using INET framework*

TTEthernet is a real-time extension of standard Ethernet protocol developed by TTTech Company in collaboration with Honeywell [11, 12]. It was standardized by SAE Company (Society of Automotive Engineers) in 2011 under the SAE AS6802 for use in avionics control systems, automotive and many other industrial applications. It fits in a transparent manner with the components of the IEEE 802.3 standard already established.

At present, the only models associated with the TTEthernet technology are provided by the project CoRE4INET (Communication over Real Time for INET framework) [11, 12]. This project is based on the extension of INET framework models of OMNeT++ to implement functions of its own elements. Its implementation requires several modifications and extensions to the core Ethernet model INET framework [11]. *Fig. 6* represents the integration of major components of the model designed in a standard INET implementation.

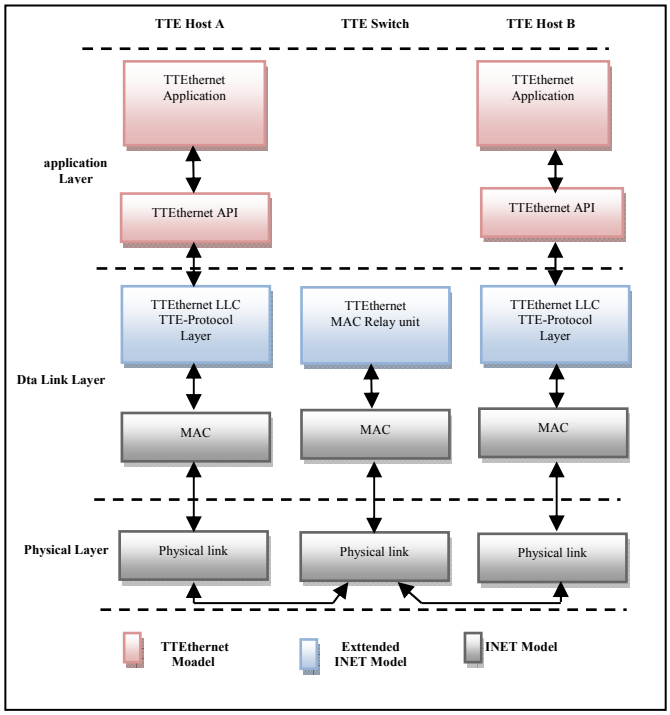

Fig. 6. TTEthernet based on INET Framework under OMNet++ [11]

Indeed, TTEthernet is a deterministic network technology and provides three communication services connected to three traffic classes with different timing requirements [11, 12]:

- The time-triggered service associated the traffic class *Time-Triggered (TT)* with the highest criticality.
- The less critical traffic is transmitted according to the best effort approach Ethernet without warranty period or reception. This traffic class is also named *Best Effort (BE)*.
- An average criticality serviced by TTEthernet. It concerns the traffic *Rate Constrained (RC)*. This service guarantees a certain bandwidth from end to end since each flow has a limit bandwidth defined by two parameters: a minimum time between two successive frames and a maximum frame size. Also, the RC traffic is statically defined by a static routing, with a single source and several receivers. These constraints are the same as those of the standard ARINC664 P7, AFDX.

So, the AFDX network supports RC traffic (or rate constrained traffic) which is a real-time deterministic traffic. Indeed, each RC message is defined by a CT-ID which is the equivalent of logic VL-ID (Virtual Link-Identifier) for the AFDX standard [11, 12]. Also, to ensure the bandwidth required for an RC message, the BAG (Bandwidth Allocation Gap-accounts) are presented as BAG-accounts that specifies the minimum time between two transmissions of consecutive frames with the same CT-ID [11, 12]. The application that sends messages RC must respect the constraints of BAG-accounts, or then the message will be invalid and rejected by the switch.

The frame format is also similar to an AFDX frame. The MAC destination address is used to identify the RC traffic. It consists of 4 bytes CT-Marker to determine the type of traffic and 2 bytes CT-ID [11, 12] to determine the message identifier (virtual link ID), as shown in *Fig. 7*.

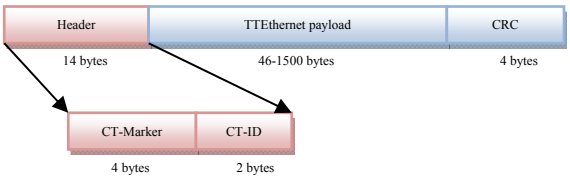

Fig. 7. TTEthernet frame format

Modeling an AFDX network based on TTEthernet model using INET framework of OMNet ++ leads to the design of three basic models: the clock model, the switch model and the terminal model (ES).

1. Clock Model

The clock, based on "ticks" is fundamental to the TTEthernet synchronization process. The time between ticks is configurable. All clocks have some inaccuracy called "clock drift" or clock deviation. These inaccuracies can fully influence the behavior of the protocol and should be supported by the model to implement.

The clock calculation formula is as following:

$$t' = t + \delta * (\Delta t_{Tick} + \Delta t_{Drift}) \quad (1)$$

With :

- t' : time of the next event,
- t : current simulation time,
- δ : planned number of ticks for a given event,
- Δt_{Tick} : duration by one tick,
- Δt_{Drift} : "*average drift*".

2. TTEthernet Terminal (End System)

Implementing a TTEthernet terminal requires the extension of Ethernet protocol with TTEthernet INET services. Therefore it incorporates a scheduler, a synchronization module and a message classification module. The mission of the scheduler is to trigger the sending and receiving messages in accordance with a messaging plan. The classification module, as the name suggests, allows classifying the received messages according to their types and process, always in accordance with the messaging plan. The AFDX ES under OMNeT++ is illustrated in *Fig. 8*.

The TTEthernet terminal can support the 3 traffics, already mentioned, TT, RC and BE. We changed this terminal and kept only the RC traffic to have an AFDX ES.

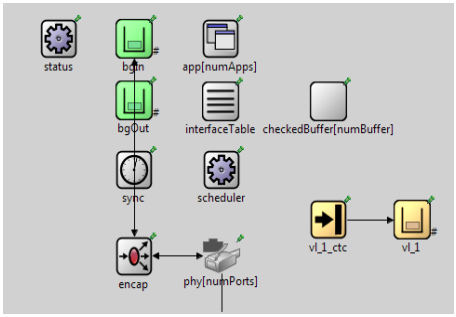

Fig. 8. AFDX END System under Omnet ++

The configuration file of each Es is in an initialization file: * .ini. The main parameters to configure, as shown in *Fig. 9*, are: (i) MAC address, (ii) VL-ID, (iii) data size, (iv) destination port and (v) BAG.

```
**.ES1.phy[*].mac.address = "0A-00-00-00-00-01"
# Defines the running real-time application of ES1 that generates RC traffic.
**.ES1.numApps = 1
**.ES1.app[0].buffers = "vl_1"
**.ES1.app[0].typename = "RCTrafficSourceApp"
**.ES1.app[0].displayName = "Vl_1"
**.ES1.app[0].payload = 500Byte
**.ES1.app[0].ct_id = 1
**.ES1.vl_1.destination_gates = "phy[0].RCin"
**.ES1.vl_1.bag = sec_to_tick(4ms)
**.ES1.vl_1.priority =          0
**.ES1.vl_1.ct_id = 1
```

Fig. 9. ES configuration file

3. *TTEthernet switch*

A TTEthernet switch must manage TT, RC and BE traffic. This is done through the extension of standard Ethernet switch. It contains a scheduler to manage the transmission plan, a local clock synchronization protocol and other protocol transmission of critical traffic [11]. The critical traffic processing module incorporates a synchronization protocol referring to the local clock described above. Its logic transmission is to classify packets according to their types based on the destination address of each of them.

On the other hand, TTEthernet switch control frames with certain constraints. They are in a preconfigured table called switch CTC-table [11]. Specifically, the control task is to check the arrival of a message at the right time and the right port. Moreover, control for a RC message is done through the verification of compliance with the BAG [11].

Similarly, we changed the TTEthernet switch and kept only the RC traffic in order to have an AFDX switch. The latter is shown in *Fig. 10.*

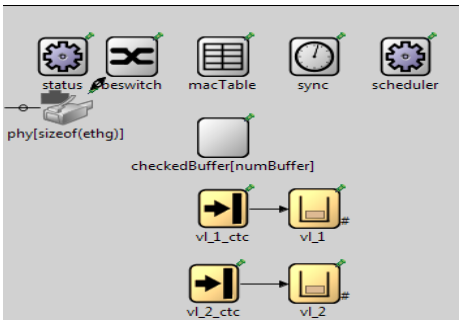

Fig. 10. AFDX switch under Omnet++

The switch configuration (receive ports, transmission port, BAG, etc.) is done in a * .ini file, as shown in *Fig. 11*.

```
**.switch1.phy[0].inControl.ct_incomings = "vl_1_ctc"
**.switch1.phy[1].inControl.ct_incomings = "vl_2_ctc"
**.switch1.vl_2_ctc.bag = sec_to_tick(1ms)
**.switch1.vl_2.destination_gates = "phy[2].RCin"
**.switch1.vl_2.bag = sec_to_tick(4ms)
**.switch1.vl_2.priority = 0
**.switch1.vl_2.ct_id = 2
**.switch1.vl_1_ctc.bag = sec_to_tick(1ms)
**.switch1.vl_1.destination_gates = "phy[2].RCin"
**.switch1.vl_1.bag = sec_to_tick(4ms)
**.switch1.vl_1.priority = 0
**.switch1.vl_1.ct_id = 1
```

Fig. 11. AFDX switch configuration file

IV. AFDX SIMULATION ON OMNET ++

A. *Case study*

In order to test and validate the AFDX model based on TTEthernet, we consider a representative example of a simple AFDX configuration network studied in the works [13] and [14]. We have chosen this example to verify and validate our AFDX simulation model by comparison to others results founded by trajectory and model checking approaches. The network is composed by three switches (S1, S2 and S3) and seven End Systems: 5 emitters (e1, e2, e3, e4 and e5) and two receivers (e6 and e7). All virtual links VL (VL1, VL2, VL3, VL4 and VL5) have a BAG = 4 ms and S_{max} = 500 octets. The entire network operates at a bitrate D=100 Mbit/s and the switching latency is L = 16 µs.

We performed the same example in OMNeT ++ using TTEthernet components with RC traffic (ES and switch). *Fig. 12* represents the considered modeled OMNeT++ topology.

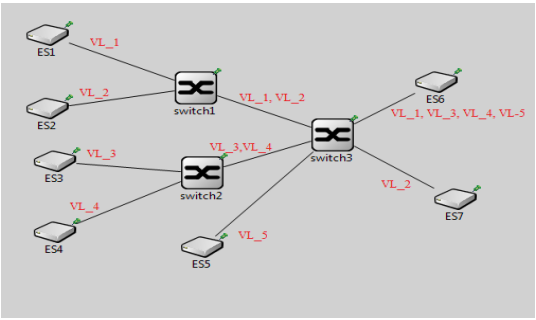

Fig. 12. AFDX case study topology

The description of the traffic across the network is shown by Table 2.

TABLE II. NETWORK TRAFIC DESCRIPTION

VL	ES source	ES destination	BAG (ms)	Smax (Bytes)	D (Mbit/s)
VL1	ES1	ES6	4	500	100
VL2	ES2	ES7	4	500	100
VL3	ES3	ES6	4	500	100
VL4	ES4	ES6	4	500	100
VL5	ES5	ES6	4	500	100

B. End to end delay analysis

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

$$D_{AFDX} = D_{ES} + (nb_l \times D_l) + (nb_{SW} \times t_{SW}) + \sum_{SW \in \{setofswitches\}} D_{SW} \quad (2)$$

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.

C. Results and discussion

We have simulated traffic described in Table 2 to determine the end to end delay of each VL. We have compared our results with the results calculated by the method of model checking (Exact worst case) as presented in [13] and those determined by the trajectory approach [14]. The results are summarized in Table 3.

TABLE III. RESULTS COMPARISON

	Paper [12]	**Paper[13]**	**Our results (simulation)**		
VL	***Exact worst case (µs)***	***Trajectory (µs)***	***Max (µs)***	***Mean (µs)***	***Min (µs)***
VL1	272	312	170 ,194	168 ,7	158, 39
VL2	192	192	212, 776	211,6	201,43
VL3	272	272	212 ,73	210,996	201,43
VL4	272	272	255,674	253 ,861	244,47
VL5	176	215	110,678	109 ,519	100 ,26

The values presented in Table 3 correspond to measurements provided by the simulator for a very specific scenario. At the S1 output port, VL1 comes always first before VL2 and at S2 port, VL3 always comes first before VL4. Otherwise at the S3 output port, VL5 appears alone and go first since it has the shortest path (ES5-S3-ES6). This explains why VL5 has the smallest latency (average latency = 109, 519 µs). VL1 and VL3 are present at the same time at the output port S3. In this case, the simulator will pass VL1 always first. VL4 comes after VL1 and VL3 at S3. This order is found in the latencies measured for each VL: average latency = 168,7 µs for VL1, VL3 with a latency of 210,996 µs and finally VL4 with a latency equal to 253,861 µs.

For VL2, the simulator gives a max value that exceed the value of the worst case determined using the method of Model Checking, since the simulator allows only the scenario where VL1 is always forwarding before VL2 at the S1 input port. Moreover, when VL1 and VL2 packets are presented at the same time, at S1, the simulator passes VL2 first, this later will have a smaller end to end delay corresponding to the time found for VL1.

In fact, VL2 will travel ES6 -Sw1- Sw3 - ES7. At the ES7 input port, it has no competition. So it will go directly without waiting. Hence, the exceedances observed compared to those obtained with the Model Checking method (Exact worst case) are justified. Therefore, the obtained values (including max values) by the OMNeT++ simulator represents a single scenario and do not take into account the worst case scenarios. This explains the lower values compared to the worst case values. Indeed, the results of OMNeT++ simulator validate the AFDX model based on TTEthernet.

V. CONCLUSION

An AFDX network simulation model is built on the OMNeT++ simulator based on the TTEthernet models of End System, Virtual link VL and Switch. To validate this model, we have chosen an illustrative example used on the literature. We have evaluated the AFDX end-to-end delay of a set of VLs. We have analyzed our results obtained by simulation on OMNet++ via comparison to others results founded by trajectory and model checking approaches. The simulation results are validated via the last comparison and the AFDX model is approved. The AFDX network design based on the OMNet++ TTEthernet model and the simulation approach can present a constructive tool and a great value in AFDX network evaluation, particularly on real-time network performance analysis and design verification.

Since, the Avionic communication architecture is an heterogeneous architecture compound of the backbone AFDX network interconnected to peripheral busses, a performance evaluation of such heterogeneous network represents a major challenge.

REFERENCES

[1] A. E. E. Committee, "Aircraft Data Network Part 7, AFDX NETWORK, ARINC Specification 664," Annapolis, Maryland: Aeronautical Radio, 2002.
[2] L. Ding, et al. "The Research of AFDX System Simulation Model", 2010 International Conf erence on Multimedia Technology,. IEEE, 2010, pp. 1-4.
[3] H. Bauer, et al., "Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach," Industrial Informatics, IEEE Transactions on, vol. 6, 2010, pp. 521-533.
[4] M. L. JOUHET, "Algorithmique du network calculus," Université de Pise, ENS de Lyon, 2012.
[5] E. Clarke, O. Grumberg, and D.A. Peled, "Model Checking" MIT Press, 1999.
[6] H. Charara et al. "Methods for bounding end-to-end delays on an AFDX network." 18th Euromicro Conference on Real-Time Systems (ECRTS'06), IEEE, 2006, pp. 10-pp.
[7] P. J. Prisaznuk, "ARINC 653 role in integrated modular avionics (IMA)," in 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, 2008, pp. 1.
[8] H. Charara, "Évaluation des performances temps réel de réseaux embarqués avioniques," 2007.

[9] D. SONG, et al., "The Modeling and the Simulation Implementation of AFDX Network System," Measurement & Control Technology, vol. 2, p. 021, 2012.
[10] J. Xia, et al., "Study on Real-Time Performance of AFDX Using OPNET," in Control, Automation and Systems Engineering (CASE), 2011 International Conference on, 2011, pp. 1-5.
[111] T. Steinbach, et al., "An extension of the OMNeT++ INET framework for simulating real-time ethernet with high accuracy," in Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, 2011, pp. 375-382.
[12] T. steinbach, et al., "real-time ethernet for automotive applications: a solution for future in-car networks," in 2011 ieee international conference on consumer electronics-berlin (icce-berlin), 2011, pp. 216-220.
[13] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an AFDX network. In Proceedings of the 18th ECRTS, Dresde, Germany, July 2006.
[14] H. Bauer, et al., "Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network," in 2009 IEEE Conference on Emerging Technologies & Factory Automation, 2009, pp. 1-8.