# Study on Real-time Performance of AFDX Using OPNET

Xia Jiqiang
School of Mechanical Engineering and Automation
Beihang University, BHU
Beijing, China
Package_l@126.com

Yuan Weimin
School of Mechanical Engineering and Automation
Beihang University, BHU
Beijing, China

Bai Ronggang
School of Mechanical Engineering and Automation
Beihang University, BHU
Beijing, China

*Abstract*—**With the upgrade of avionics system integration, the requirements to huge throughput, low latency and high reliability of communication network have become so urgent, that the traditional Aircraft Data Networks (ADN) cannot meet the requirements. In this paper, a brief analysis to the next-generation ADN protocol--avionics full duplex switched Ethernet (AFDX) and construction of end system and AFDX switch are presented, and then, based on OPNET platform, a typical simulation of AFDX network system is given for its real-time performance.**

***Keywords-AFDX; simulation; end-to-end delay; Virtual Links***

## I. INTRODUCTION

With the strict requirements of flight safety in various flight conditions and entertainment, in the current aircraft, the integrated functions number and the avionics embedded systems are becoming huge and complex separately. These means increasing of quantity of data exchanged and thus the increasing number of connections between functions.

To solve this, several avionics architectures were developed based on the concept of modular architecture. But the most of them resort to rather old means of communication, like the ARINC 429 data buses which are mono transmitter buses and with limited performances (100 kbps) [1]. This leads to limited Bandwith and a huge number of buses, obviously it is not acceptable in terms of weight and wiring complexity. The solution proposed by Boeing was to embed ARINC 629 data buses which are multi transmitter buses with much better performance (2 Mbps up to 120 users) [2]. However, this technology is not considered any more as a satisfactory solution because of the important development costs of the specific components. For this reason, Boeing and Airbus both turn to commercial ethernet technology. The disadvantage of Ethernet, opposite to an avionics application, is the non intrinsic determinism of its access method to the physical support, CSMA/CD (which induces possible collisions on the point-to-point links level).

In order to cope with this problem, the AFDX (Avionics Full Duplex Switched Ethernet, AR-INC 664) used for modern aircraft such as Airbus A380 was defined , and representing a major upgrade in both bandwidth and capability for aircraft data networks, AFDX has become the reference communication technology in the context of avionics. AFDX is a full duplex switched Ethernet network to which new mechanisms have been added in order to guarantee the determinism of avionic communications.

But, the ARINC 664 shifts in fact the problem to the switch level where various flows will enter in competition for the use of the resources of the switch, and then it is necessary to study the performance of the network.

Thus, this paper is mainly concerned evaluating the end-to-end transfer delay through the network.

## II. OVERVIEW OF THE AFDX PROTOCOL

AFDX is such a communication protocol that, based on the Commercial Ethernet Stand, adopts most of the IEEE802.3 and UDP/IP protocols, and is optimized according to the integration of the avionics sub-system. In accordance with the hierarchical division, this protocol includes four layers, the Transport Layer, Network Layer, Data Link Layer, and Physical Layer(see Figure 1) [3].

When an application sends a message from the source sub-system to the application of the destination one, the source End System, the AFDX switches, and the destination End Systems are configured to deliver the message appropriately by the ports. That is to say, as main parts of the AFDX network, the end system and the AFDX switch are loaded the protocol stack. This can be illustrated by the small example of figure 2.

A message M is sent to Port S1 by the Avionics subsystem. END System 1 encapsulates the message in an Ethernet frame and sends the Ethernet frame to the AFDX Switched Network on Virtual Link 101 (the Ethernet destination address specifies VLID 101). The forwarding tables in the network switches are configured to deliver the Ethernet frame to End System 2. The End Systems that receive the Ethernet frame are configured so that they are able to determine the destination ports for the message contained in the Ethernet frame. In the case shown in Figure 2, the message is delivered by End System 2 to port D.

978-1-4577-0860-2/11/$26.00 ©2011 IEEE

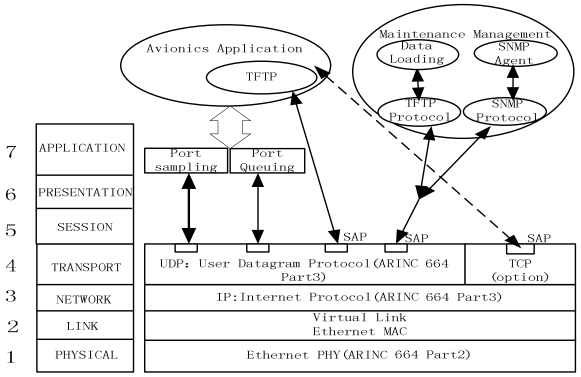

Figure 1. The AFDX Protocol Stack

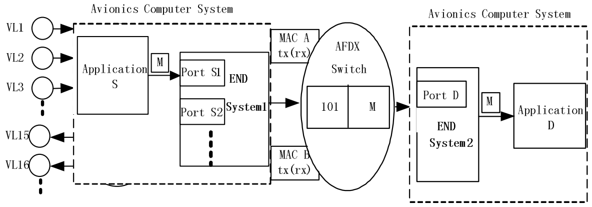

Figure 2. The AFDX Message Flow

### A. Virtual Links

The central feature of an AFDX network is its Virtual Links (VL). In one abstraction, it is possible to visualize the VLs as an ARINC 429 style network each with one source and one or more destinations. Virtual Links are unidirectional logic path from the source end-system to all of the destination end-systems. Unlike that of a traditional Ethernet switch which switches frames based on the Ethernet destination or MAC address, AFDX routes packets using a Virtual Link ID. Contrasting to the 48 bits totally uncertain MAC address of ethernet frame, the Virtual Link ID is a 16-bit unsigned integer value that follows the constant 32-bit field. The switches are designed to route an incoming frame from one, and only one, End System to a predetermined set of End Systems [4].

The End-system communication stack should guarantee in transmission the allocated bandwidth of each Virtual Link regardless of the attempted use of Bandwidth by other Virtual Links, in order to preserve segregation between partitions at the network level. One Virtual Link should not be shared by two or more source partitions.

#### 1) parameters

The main characters of AFDX network are determinacy and redundancy. To guarantee these, just as partitions are used to isolate Avionics subsystems from one another, a similar mechanism is required to isolate individual virtual links, in order to prevent the traffic on one virtual link from interfering with traffic on other virtual links using the same physical link This is done by limiting the rate at which Ethernet frames can be transmitted on a virtual link and by limiting the size of the Ethernet frames that can be transmitted on a virtual link.

Each virtual link is assigned two parameters:

- Bandwidth Allocation Gap (BAG), a value ranging in powers of 2 from 1 to 128 milliseconds
- Lmax, the largest Ethernet frame, in bytes, that can be transmitted on the virtual link

The BAG represents the minimum interval in milliseconds between Ethernet frames that are transmitted on the virtual link. For example, if a virtual link with VLID 1 has a BAG of 32 milliseconds, then Ethernet packets are never sent faster than one packet every 32 milliseconds on VLID1. The BAG values should comply with the formula [5]:

$$BAG = 2^k [in\ ms], \text{(k integer in range 0 to 7).}$$

The choice of BAG for a particular virtual link depends on the requirements of the AFDX ports that are being provided link-level transport by the virtual link. If VLID 1 has an Lmax of 200 bytes, then the maximum bandwidth on VLID 1 is 50,000 bits per second (200* 8*1000/32).

#### 2) scheduling

Virtual Link scheduling consists of two components: packet regulation and multiplexing.

Though there may be one or more ports connected to a VL, and many VLs exist in an End System, the traffic regulation must be based on per VL. On a per VL basis the traffic regulator (traffic shaping function) should shape the flow to send no more than one frame in each interval of BAG milliseconds (i.e., create zero-jitter output streams). A frame will not be transmitted while a VL is eligible but has no data for transmission.

There are usually several VLs within an End System, and then it is necessary to determine the transmission sequence of these output streams. In fact, this function is fulfilled by the scheduler.

The outputs of the Regulator consist of regulated streams of Ethernet frames. The Virtual Link Scheduler is responsible for multiplexing the regulator outputs into the Redundancy Management Unit for replication and transmission on the physical links. Fig.3 shows the Virtual Link scheduling scenario.

Jitter is introduced when the Regulator outputs are combined by the Virtual Link Scheduler MUX; Ethernet frames arriving at input to the MUX at the same time will experience queuing delay (jitter). Not only must the Virtual Link Scheduler ensure the BAG and Lmax limitations for each virtual link, but it is also responsible for multiplexing all of the virtual link transmissions so that the amount of jitter introduced by the multiplexing is within acceptable bounds. The ARINC 664 specification requires that, in transmission, the maximum allowed jitter on each virtual link at the output of the End System comply with both of the following formulas [5]:

$$max\_jitter \le 40\mu s + \frac{\sum_{j\in\{set\ of\ VLs\}} (20 + Lmax_j) \times 8}{Nbw}$$

$$max\_jitter \le 500\mu s$$

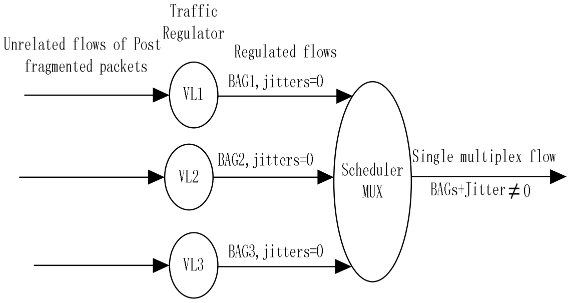

Figure 3. Virtual Link scheduling scenario

Nbw is the link bandwidth (100 Mbps). The first formula represents a bound on the jitter arising from an Ethernet frame being delayed by a frame from each of the other virtual links. The second formula is a hard limit, independent of the number of virtual links. These requirements are necessary to demonstrate the overall "determinism" of an AFDX network.

### B. Redundancy Management

The other most important character of AFDX that differs from the Standard Switch Ethernet is the redundancy management.

In AFDX, each frame from sending End Systme will be forwarded to receiving End System by two independent network [7].

Transmitting End System receives data from avionic subsystem partition, and through the communications protocol stack, a sequence number that increases by one is added to the frame as part of the MAC payload, just before the FCS field. In according to this number, only can the receiving ES reconstruct a single ordered stream of frames.

In default mode each frame is sent across both of two networks. Upon reception, an algorithm in the communications stack (below IP layer) uses a "First Valid wins" policy. This means that the first frame to be received from either network with the next valid sequence number is accepted and passed up the stack to the receiving partition. When the second frame is received with this sequence number, it is simply discarded.

Redundancy management can solve the problems effectively of abnormal frames, loss of frames and stuck frames. (See Fig. 4)

It should be noted that the problems illustrated in Fig. 4 occur seldom and impossibly appear in the same network (network A as Fig. 4) in an actual AFDX network.

## III. MODELING

OPNET Modeler is powerful for network simulation. It is commercial software distributed by the OPNET Technologies Ins., which is capable of analyzing the behavior and performance of complex network. In this environment, data collection and statistic are fulfilled via inserting standard or user-defined probes in any position of the network model.

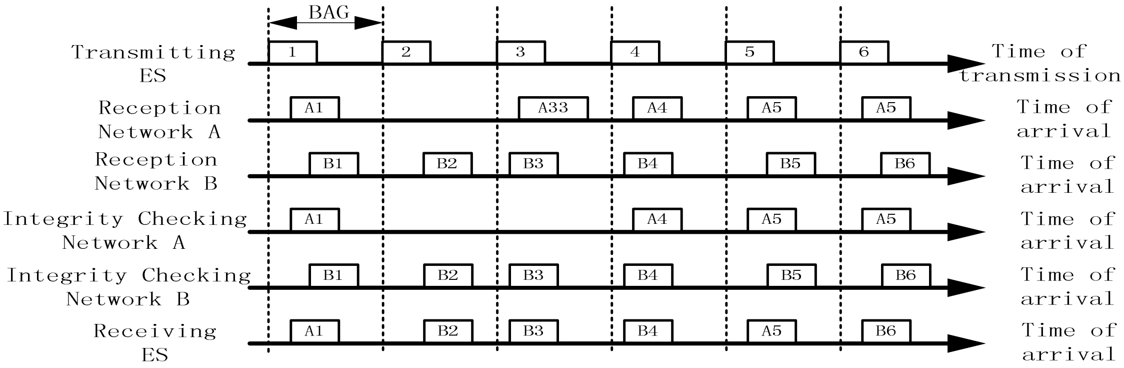

Figure 4. Redundancy Management

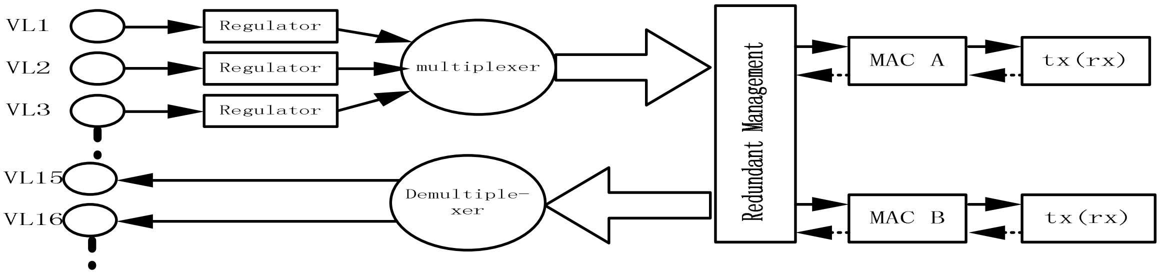

Figure 5. End System Mode

Note that this study is concerned with the delay of a frame crossing the network. The amount of the time taken during the communicating of the avionics sub-system and End System is beyond this study. In addition, we will determine the queue size (the maximum and the distribution) of the output port of a switch, where there is a congestion problem [8].

### A. *ES Modeling*

A source or destination ES should complete the functions of receiving from or transmitting to an avionics sub-system via a port, processing through the protocol stack, nevertheless, according the foregoing emphasis, the ES is modeled as presented in Fig.5 [9]

The End System model, as we can see in Figure 6, is made up of three types of modules: message generation module, result collection module and protocol control module. The Sx module is responsible of generating messages and collecting the results, the other processer modules or queue modules fulfill the function of protocol stack.

### B. *Switch Modeling*

Model of switch is composed of input queues, switch core module and output queues. We depict the model of Switch in Figure 7.

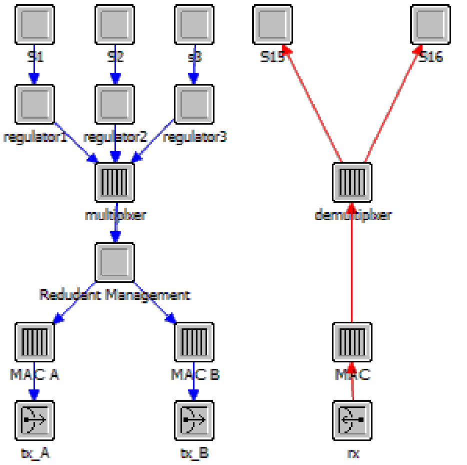

Figure 6. ES node model in OPNET modeler

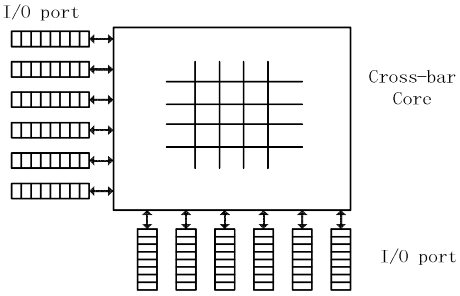

Figure 7. Switch Model

Input queue consists of filtering and policing function blocks. If the frame properties do not comply with the configuration parameters, the frame is filtered (i.e. discarded). These functions apply test rules about frame integrity, frame length, traffic budget and acceptable destination. The role of the switch core module is to route frames toward the output, according to a static routing table.

In OPNET, we choose a library model of ethernet switch. According to the characters of AFDX switch, we modify the process models and reconfigure the relative parameters based the chose model.

## IV. SIMULATION SCENARIOS AND RESULTS

### A. *the simulation of typical AFDX network*

#### 1) *the Configuration of End System and Network Topology*

According to a test configuration provided by Airbus for an industrial research study [10], a typical AFDX can be characterized like this: the End System, as the inputs and outputs are connected by several interconnected switches (like 8 switches), there are no buffers on input ports and one FIFO buffer for each output port. Each End System is connected to exactly one switch port and each switch port is connected to at most one End System. Links between switches are all full duplex. Number of input and output End Systems per switch are not specified, as is shown in Fig. 8.

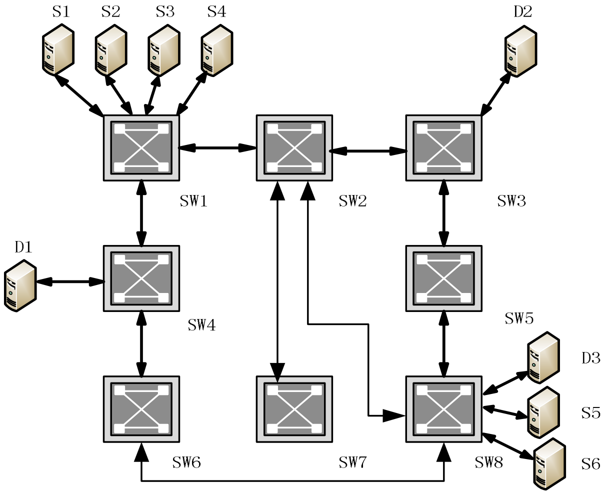

Figure 8. A Typical AFDX Network Topology

To be consistent with the smallest buffer queue in AFDX protocol, the input buffer queue and output port queue are configured separately 3036 bytes and 256 greatest data frames length [6]. In addition, to simplify the simulation, each VL only supports one sub-VL.

#### 2) *the Analysis of the Simulation Result*

Figure 9 depicts end to end delay distribution of three VLs, VL1 (S1->SW1->SW2->SW8->D3, BAG=4ms), VL5 (S5->SW8->SW6->SW4->D1, BAG=128ms), and VL6 (S6->SW8->SW5->SW3->D2, BAG=16ms).

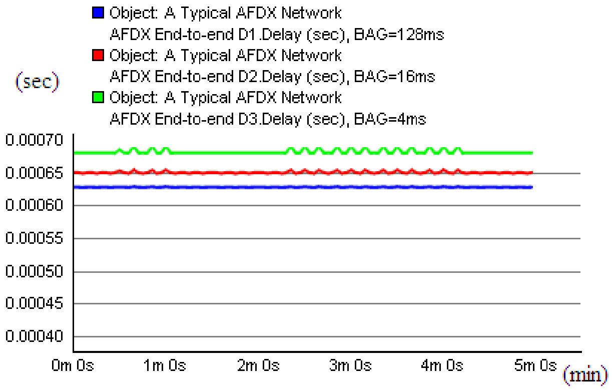

Figure 9. The End-to-end Delay

Analyzing Fig. 9, we can draw the following conclusions:

a) The AFDX end-to-end delay ranges from 0.6 ms to 0.9 ms. According to the data proposed in [11], the analytical upper bound in a AFDX network with the same configuration is 1.062ms, this comparison validate our proposed network model to get a realistic AFDX network model. Meantime, the different VLs end-to-end delay is in definite scope [5], which meets the ARINC664 standard.

b) The VLs with different BAG have different delay performance. The total rule is that the delay of VL with smaller BAG is larger. The reason is that the load of the network with smaller BAG is often bigger, thus AFDX Switch needs more time to process, while the total AFDX network delay depends mainly on queuing delay in AFDX Switch.

## V. Conclusions

In the light of the characteristic of industrial control network, the real-time research of AFDX is presented. Following the brief introduction of key technology, the end-to-end delay of a typical AFDX network is simulated on the OPNET platform, based on the end system, virtual links and AFDX switch. The simulation result validates the model. Because of the promotion effect of simulation--reducing development cost and shortening development cycle in the process of development, the model proposed in this paper will provide a useful tool for the promotion of the AFDX protocol.

## Acknowledgment

The research in this paper is supported by the Chinese National Programs for High Technology Research and Development (863 program) and the grant number is 2008AA040207.

## References

[1] ARINC 429, Aeronautical Radio Inc. ARINC specification 429. Digital Information Transfer System (DITS) parts 1,2,3., 2001.

[2] Jean-Luc Scharbarg and Christian Fraboul, “Simulation for end-to-end delays distribution on a switched Ethernet,” Emerging Technologies and Factory Automation(ETFA 2007), IEEE Conference on, Sept. 2007, pp. 1092, doi: 10.1109/EFTA. 2007.4416904.

[3] ARINC664, Aeronautical Radio Inc. ARINC specification 664. AIRCRAFT DATA NETWORK (ADN)part7, 2005

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

[5] Ye Jia-zi, Chen Xiao-gang and Zhang Xin-jia, “Design of Av ion ics Commun ica tion Network Ba sed on AFDX,” Journal of Measurement & Control Technology, Vol. 27, Num. 6, 2008, pp. 56-60, doi: CNKI:SUN:IKJS.0.2008-06-020.

[6] Song Dong, Zeng Xingxing, Ding Lina and Hu Qiong, “The Design and Implementation of the AFDX Network Simulation System,” Multimedia Technology (ICMT, 2010), IEEE Conference on, Oct. 2010, pp. 1, doi:10.1109/ICMULT.2010.5629728.

[7] Niu Bing, “Hardware Design of AFDX Switch and Scheduling of End System,” XIDIAN UNIVERSITY, Jan. 2009, doi:CNKI:CDMD:2.2010.082720.

[8] Charara, H. and Fraboul, C., “Modelling and Simulation of an Avionics Full Duplex Switched Ethernet”, Telecommunications, 2005. advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop. aict/sapir/elete 2005. proceedings, July 2005, pp. 207, doi:10.1109/AICT.2005.58.

[9] Ding Lina, Song Dong, Zeng Xingxing and Hu Qiong, “The research of AFDX system simulation model,” Multimedia Technology (ICMT), 2010 International Conference on, Oct. 2010, pp. 1, doi: 10.1109/ICMULT.2010.5632133.

[10] 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.

[11] Scharbarg, J.-L., Ridouard, F. and Fraboul, C., “A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network,” Industrial Informatics, IEEE Transactions on, Feb. 2009, pp. 38, doi: 10.1109/TII.2009.2016085