Impact of End System Scheduling Policies on AFDX Performance in Avionic On-Board Data Network

Chakkaphong Suthaputchakun

Electrical and Electronic Engineering Department
Bangkok University
Thailand

E-mail: chakkaphong.s@bu.ac.th

Kin-Man-Benjamin Lee, Zhili Sun

Institute for Communication Systems
University of Surrey
UK

E-mail: 6154294@surrey.ac.uk, z.sun@surrey.ac.uk

Abstract— Ordinary aircrafts rely on point to point wire connection to transmit data. These wires add additional weight to the aircrafts and thus, the fuel cost is increased. Aircrafts released in recent years used AFDX protocol to transfer data within the aircraft. AFDX is a deterministic network transfer protocol used in aircrafts to ensure the quality of service (QoS) on the network and reduce the wiring needed. However, the specification of AFDX only defines the required performance without providing the methods to achieve it and hence there is a room for research. The motivation of this paper is to investigate and analyse impact of different scheduling policies of End System on the performance of a real avionic on-board data network running AFDX protocol.

Keywords— *AFDX, Avionic network, scheduling policy, smallest BAG, round robin, longest queue, smallest size, FIFO*

I. INTRODUCTION

Since the requirements of the data communication on Aircrafts increase, it is necessary to improve the data transmutation rate as well as the quality of service (QoS). The latest standard is proposed by Aeronautical Radio Incorporated (ARINC) for avionic systems called Avionics Full-Duplex Switched Ethernet (AFDX) or formally the ARINC 664 part 7 [1-4]. However, this AFDX standard only defines the performance of the system required without the method to achieve the goal. Hence, there are rooms for research.

The aim of this paper is to analyse the performance of the AFDX network on the simulation software called OMNeT++ [4]. AFDX network elements simulation module, namely AFDX End Systems (End System), AFDX Switches (Switches) and AFDX Virtual Links (Virtual Links) are developed within this paper carrying different functionalities on transmitting, scheduling and receiving the AFDX frame which is the data packet to be transmitted within the network.

The paper is organized as follows. An overview of AFDX standard is given in Section II. Section III provides detail of developed simulation modules for AFDX communication, while Section IV describes detail of the system model. The performance evaluation results are discussed in Section V and finally Section VI concludes the paper and provides guidelines for future research.

II. AFDX OVERVIEW

This section covers the overview of AFDX and the advantages of using AFDX [5-9]. The structure of AFDX elements, including AFDX frame, AFDX End System, AFDX Switch and Virtual Link and how they are linked together to provide a deterministic QoS are discussed in this section.

A. Introduction

After the success of Ethernet in commercial computers in the past decade, there is a tendency to transfer the Ethernet technology into different fields of communication system [6]. Avionics Full-Duplex Switch Ethernet (AFDX) is one of them, using the basis of commercial Ethernet and improving the reliability and real-time requirement of Avionics systems. The AFDX network is designed to replace the existing avionic network data bus module ARINC 429 which was developed in 1977 [7].

The specification of AFDX is specified in ARINC664-part 7 standard which is standardized by Aeronautical Radio Incorporated (ARINC). The standard stated the protocol requirements, electrical requirements and the communication channels between different avionic systems. By using the ARINC standard, AFDX is developed to provide a high speed (100Mbps), real-time and fault-tolerance deterministic network. This system stability is guaranteed by a limited bandwidth on each transmission link and data packets are sent in two redundant networks to ensure data delivery. Besides, the system provides a maximum delay measurement called jitter that indicated the maximum delay of the data packet in network due to scheduling. The AFDX network consists of three main network components, namely End System and Switch [8].

B. AFDX End System

The design of AFDX End System (ES) is to provide service of guaranteeing secure and reliable data exchange amongst the partition software. Each avionic computer system is equipped with an End System as its network interface through which the host applications on the avionic system, i.e. avionic subsystems, can communicate with other avionic computer systems. Figure 1 illustrates the conceptual design of an AFDX network.

Fig. 1 Example of AFDX network with redundancy connection

In Figure 1F, different host applications on an avionics system (e.g. Avionics Computer A) send their application data to the connected avionics subsystem partitions, which further deliver the data to End System for centrally processing. End System is responsible for packaging and sending out the received data to the connected AFDX Switch, which later dispatches packets toward the destination avionics computer system (e.g. Avionics Computer B). If redundancy management is enabled, the communication between two Avionics systems will experience two AFDX Switches, AFDX Switch 1 and AFDX Switch 2 (for redundancy purpose). Upon receipt of data packet (or the first received valid packet if redundancy management is enabled), the destination End System would report to corresponding avionics subsystems. In order to achieve deterministic communications inside an AFDX network, End System and Switch are designed with following features: Virtual Link and MAC Addressing, Flow/Traffic Control and Scheduling.

1) Virtual Link

The concept of Virtual Link is used in an AFDX network to build virtual connections for both sides of communications. The virtual links isolate the underlying available bandwidth of physical connection, which therefore provide feasibility for a host application to allocate multiple communication channels. The mechanism of isolation is also helpful in protecting individual Virtual Link from being affected by other Virtual Links sharing the same physical bandwidth. Figure 2 depicts the concept of virtual link(s).

Fig. 2 Virtual link for AFDX End System

As illustrated in Figure 2, a Virtual Link is a logical unidirectional connection which originates from a source End System to one or more destination End Systems. As specified in the AFDX standard, an End System can be designed to only receive VLs and not transmit VLs, or only transmit VLs and not receive VLs, or transmit and receive VLs. However, it

should be assured that any VL can originate from one and only one End System.

2) Algorithm of Flow & Traffic Control and Scheduling

Two parameters are defined to regulate the traffic carried by VL(s): Bandwidth Allocation Gap (BAG) and jitter. BAG specifies the minimum time interval between the first bits of two consecutive frames on the same VL. Figure 3 shows an example of traffic regulation on a Virtual Link.

Fig. 3 Regulator input and output

In Figure 3, if the data flow inputted into a regulator is unregulated, the possible consequence could be congestion on an End System's output port, especially when a Virtual Link is under heavy load. The unpredictable arrival of data packets could lead to inefficient bandwidth usage. The idea of BAG is to shape data flow by defining minimum timeslot between two consecutive frames. Specifically, only one data frame is allowed to be processed within each slot, and the processing should start from the beginning of a slot (illustrated as "Regulator Output" in Figure 4), by which the performance of network communication is possible to be predicted and measured. BAG is a value ranging in power of 2 from 0 to 7 (i.e. from 1 to 128 milliseconds).

As mentioned before, an End System could be featured with multiple Virtual Links. A specific virtual link scheduler is needed to multiplex data frames from different Virtual Links onto the same physical Ethernet link (as shown in Figure 4).

Fig 4 Virtual link scheduler

In Figure 4, multiple regulated traffics are fed into a unified scheduler. Jitter will not be present when traffic is processed by traffic regulator which works on per-link basis. However certain jitter is measurable on multiplexed flow due to the process delay on the Scheduler MUX. AFDX standard specifically defines the concept of Jitter, which is used to bound the upper limit of transmit latency between the start of a BAG and the first sent bit of the frame being transmitted within the corresponding slot.

Jitter is applied at the output of an End System, i.e. the output of a scheduler, to measure the contention of data frames when a scheduler is scheduling VLs (Figure 5).

Fig.5 Virtual Link scheduling with jitter

The range of jitter for each VL at the output of an End System is defined as follows:

$$\max_jitter \leq 40\mu s + \frac{\sum_{j \in \{set of VLs\}} (20bytes + L_{max_j}) \times 8}{Nb w} \quad (1)$$

$$\max_jitter \leq 500\mu s \quad (2)$$

where L_{max_j} denotes the maximum allowed frame size on the j^{th} Virtual Link, Nbw is the link bandwidth in Mbps (e.g. 10 Mbps, 100 Mbps). Equation (1) is for an end system having a small number of Virtual Links with small frame sizes. Equation (2) is the upper limit for all cases of Virtual Links on an end system.

Each Virtual Link must be assigned with a certain BAG and L_{max} . The selection of BAG and L_{max} for a Virtual Link depends on the designed traffic load. For example, if a Virtual Link is asked to serve two communication ports whose communication frequencies are 10 Hz and 40 Hz, an average communication frequency on this link will be 25 Hz equivalent to 40 ms. In order to ensure that data packets are properly processed provided with sufficient bandwidth, BAG value should be less than 40 ms. The first available BAG is 32 ms.

3) Redundancy Management

Redundancy management in an AFDX network is achieved through employing independent and redundant networks. If redundancy management is enabled, each End System connects to two networks, i.e. network A and B, and send data packet through these two networks respectively. A sequence number is added to each frame and increase on each successive frame. Upon receipt of data frames, the receiving End System would check the frame with the policy of “First Valid wins”, which means the first valid frame received by the destination End System, will be accepted. The secondly received or invalid frame will be simply discarded.

C. AFDX Switch

AFDX switch is the device that interconnects both communicating End Systems, and polices data traffic according to the specified configurations. There are five functional blocks defined for an AFDX switch, Filtering & Policing Function, Switching Function, End System, Configuration Tables and Monitoring Function (as shown in Figure 6).

Fig. 6 AFDX switch function blocks

- **Filtering & Policing Function:** All frames arriving at a switch will be firstly examined by this function in terms of frame integrity, frame length, traffic budget, validity of destination, etc.
- **Switching Function:** As the core function of a switch, Switching function is responsible for forwarding data packets, which are filtered through the Filtering & Policing Function with success, toward appropriate output ports.
- **Configuration Tables:** Configuration tables contain static configuration data used to control the Filtering & Policing function and Switching function.
- **End System:** The end system of a switch is for implementing communications amongst switches, e.g. receiving data packets dedicated to a switch, sending relevant management data from a switch.
- **Monitoring Function:** The monitoring function cooperates together with all other functions to monitor any necessary events, e.g. event log, statics of internal situation.

Fig. 7 Partition module

III. AFDX SIMULATION MODULES

This section shows the AFDX simulation modules created in the paper. There are 3 main simulation modules, which consists of Partition, End System, and Switch modules.

A. Partition Module

Partition module is computer partition connected to the End System. It acts as sources and destinations of data in the

AFDX network. The Partition module is shown in Figure 7. The numbers of Message-Source and Message-Dest in each Partition will be dynamically set at a beginning of simulations.

B. End System Module

The End System module shown in Figure 8 consists of several sub-modules including traffic shaping, scheduling, and redundancy management for data transmission and reception of each Virtual Link.

Fig. 8 End System module

C. Switch Design

The Switch module is illustrated in Figure 9. There are two main compound modules in the Switch module, which are Switch-Port and Switch-Fabric. Switch-Port acts as an interface for data transmission and reception and filters all received frames, while Switch-Fabric performs routing and scheduling all received frames to the appropriated ports. It is noted that all of the connections in the Switch are full duplex.

Fig. 9 Switch module

D. AFDX Network Configuration File

The configuration file supplied from the industry is in eXtensible Markup Language (XML) format. Basically, the file contains all essential information of End Systems, Switches and Partitions that linked within each End System. Due to page limitation, we cannot provide the schema of the configuration file in the paper.

OMNeT++ [10] does not directly support XML file parsing. Therefore, the XML file parsing library, called RapidXml, has been used for file parsing. Then, Net-Builder which is a file parser will parse the AFDX configuration file and dynamically create simulation module in OMNeT++. Figure 10 shows a sample of the configuration of a transmission and a reception Virtual Link in an End System. The txVirtualLink node and rxVirtualLink node are defined in the layer "Configuration/endSystem/".

```
<txVirtualLink vlid="3">
  <networkSelector>AB</networkSelector>
  <bag>128000</bag>
  <sizeMax>64</sizeMax>
  <numberSubVl>1</numberSubVl>
</txVirtualLink>
<rxVirtualLink vlid="1">
  <configuration>RI</configuration>
  <bag>128000</bag>
  <sizeMax>161</sizeMax>
  <skewMax>150.0</skewMax>
</rxVirtualLink>
```

Fig. 10 Configuration of Virtual Links in an End System

Fig. 11 Use case network topology with the switch port identifiers

IV. SYSYEM MODEL

The use case system model, which is the reference existing topology from an airplane industry, is shown in Figure 11. The model focuses on the payload and transforms the current SpaceWire SSMM switching network into a centric AFDX network with IO module added to manage low bit rate instruments and 1553 sub network.

The use case simulation scenario is comprised of 18 end systems, which consist of 6 instruments (Equipment 1-6), 8 platform equipment (SSMM, PCDU 1, PCDU 2, EPS, STR 1, STR 2, STR 3, and RIU), 2 I/O modules (IO M 1, and IO M 2), and 2 On Board Computers (OBC 1, and OBC 2).

The simulation scenario also consists of 3 switches in redundant. The network topology of 3 switches aims to provide alternative physical paths for the same virtual link. These switches ensure the switch policy defined in the ARINC

664 P7 switching specification. All the AFDX end systems and switches are redundant as presented with the white squares in the figure.

This detailed configuration of the first five Virtual Links out of ninety-five in the AFDX network is shown in Table I.

TABLE I: THE AFDX NETWORK VIRTUAL LINK KEY CONFIGURATIONS

Virtual Link ID	BAG (s)	Freq. (Hz)	Size (byte)	Source	Dest.
10	0.064	8	251	OBC1	IO_M_2
20	0.016	32	1518	IO_M_1	SSMM, OBC2
30	0.064	8	299	IO_M_2	OBC1, OBC2
40	0.064	8	333	OBC1	EQUIPMEN T5
50	0.064	8	121	EQUIPMEN T2	SSMM, EQUIPMEN T1, EQUIPMEN T3, EQUIPMEN T4, EQUIPMEN T5, EQUIPMEN T6

V. PERFORMANCE EVALUATION RESULTS

In order to study the performance of AFDX in avionic on-board data network, AFDX modules and sub modules have been developed and tested on OMNeT++. The paper studies an impact of End System scheduling policies on the performance of the AFDX avionic on-board communication by varying the scheduling policies on the End System to Round Robin, Smallest BAG, Longest Queue, Smallest Size, and FIFO, respectively. The detail of each individual scheduling policy is described as follow.

Round Robin: Round Robin policy allows the data frames of all queues, i.e. all virtual links in this case, to be regulated in each turn. During each round of the operation, all virtual links are guaranteed to be equally served one by one. The aim of Round Robin is to provide fairness of all virtual links regardless their arrival time, queue size, and etc.

Smallest BAG: Smallest BAG policy will allow the data frame of the virtual link of which the BAG value is the smallest to be regulated before the other data frame regardless their arrival time at the queue.

Longest Queue: Longest Queue policy aims to reduce the queue size as quickly as possible to avoid frame dropping due to no space left in the queue for a new coming frame. This policy allows a data frame of the virtual link, of which the number of frames waiting in the queue for traffic shaping is the largest, to be regulated before the others. This policy hence is suitable for the dense traffic communication where the frame dropping rate is high.

Smallest Size: Smallest Size policy allows the smallest data frame to be regulated before the other larger data frame regardless their arrival time at the queue.

FIFO: First-In-First-Out scheduling policy is the simple and well-known algorithm. The FIFO allows the data frame to be regulated according to their arrival time at the queue. The data frame which arrives first will be regulated before the others.

A. Average End System Jitter

The performance evaluation results in term of the average ES jitter with different scheduling policies is shown in Figure 12. From this figure, it is observed that the Round Robin achieves the smallest average ES jitter compared to the others, while the Smallest BAG and Smallest Size policies experience the worst ES jitter.

Fig. 12 Performance evaluation results in term of the average ES jitter with different scheduling policies

B. Maximum End System Jitter

Figure 13 shows the performance comparison in term of the maximum ES jitter in the AFDX network with different ES scheduling policies. In contrast to the previous section, Round Robin experiences the largest maximum ES jitter, while Longest Queue and FIFO attain the lowest maximum ES jitter.

Fig. 13 Performance evaluation results in term of the maximum ES jitter with different scheduling policies

C. Number of Data Frames facing larger than 500μs Jitter

The last performance metric is the performance evaluation in term of the percentage of the number of data frames that

experience larger than 500us ES jitter which is the maximum ES jitter recommended by the AFDX standard. It can be seen from Figure 14 that Longest Queue and FIFO achieve the lowest percentage in term of the number of data frames facing larger than 500us ES jitter compared to the others. Round Robin achieves the bit worst compared to the Longest Queue and FIFO, while the Smallest BAG and Smallest Size are the worst two policies so far.

Fig. 14 Performance evaluation results in term of the percentage of the number of data frames that experience larger than 500us ES jitter with different scheduling policies

Therefore, from these performance evaluation results, it is recommend implementing Longest Queue or FIFO as ES scheduling policy in order to achieve the optimal performance in term of End System jitter. These two schedulers outperform the others by avoiding a chance that data frames are stuck in transmission queues for long time. Therefore, most data frames experience almost the same jitter, and hence it reduces the maximum ES jitter value and the number of data frames facing larger than 500us jitter while giving reasonably low average ES jitter compared to the other schedulers.

VI. CONCLUSION

This paper evaluates the performance of AFDX in avionic on-board data network based on different scheduling policies implemented on End System. The performance is measured in

term of End System jitter. The results show that Longest Queue and FIFO schedulers outperform the others by giving low ES jitter with lowest number of delayed data frames. Therefore, the Longest Queue or FIFO is recommended as End System scheduler for AFDX on-board network.

Since the current AFDX network has a low network load, for the future work, the network can be loaded with more traffic if needed to utilize the AFDX network. In addition, developments of other scheduling algorithms are also recommended to improve the overall performance of the AFDX network in the future.

REFERENCES

- [1] Std-A651 ARINC 651-1 Design Guidance for Integrated Modular Avionics
- [2] Std-A653/1 ARINC 653P1-2 Avionics Application Software Standard Interface, Part 1, Required Services
- [3] Std-A653/2 ARINC 653P2 Avionics Application Software Standard Interface, Part 2, Extended Services
- [4] Std-A664/7 ARINC_664P7-1_2009-09-23 : Aircraft data network part 7 AFDX
- [5] Meng Li; Lauer, M.; Guchuan Zhu; Savaria, Y., "Determinism Enhancement of AFDX Networks via Frame Insertion and Sub-Virtual Link Aggregation," *Industrial Informatics, IEEE Transactions on* , vol.10, no.3, pp.1684,1695, Aug. 2014
- [6] Brajou, F.; Ricco, P., "AFDX-based flight test computer concept," *Instrumentation & Measurement Magazine, IEEE* , vol.8, no.3, pp.55,58, Aug. 2005
- [7] Yao, M.; Qiu, Z.; Kwak, K., "Leaky bucket algorithms in AFDX," *Electronics Letters* , vol.45, no.11, pp.543,545, May 2009
- [8] Yu Hua; Xue Liu, "Scheduling Heterogeneous Flows with Delay-Aware Deduplication for Avionics Applications," *Parallel and Distributed Systems, IEEE Transactions on* , vol.23, no.9, pp.1790,1802, Sept. 2012
- [9] Bisson, K.; Troshynski, T., "Switched Ethernet testing for avionics applications," *Aerospace and Electronic Systems Magazine, IEEE* , vol.19, no.5, pp.31,35, May 2004
- [10] OMNeT++ Network Simulation Framework. [Online]. Available: <http://www.omnetpp.org>.