Comparative Analysis of Next Generation Aircraft Data Networks

Osman Rasit Kultur
Avionics Hardware Design Department
Aselsan Inc.
Ankara, Turkey
orkultur@aselsan.com.tr

Hasan Sakir Bilge
Electrical and Electronics Engineering
Gazi University
Ankara, Turkey
bilge@gazi.edu.tr

***Abstract*—Aircraft have many sensors and modules scattered throughout their fuselage. Due to the ever-increasing sensor load (radar, electronic warfare, radio, camera, etc.), aircraft need more bandwidth and higher broadband communication backbones. Legacy protocols such as ARINC-429 and MIL-STD-1553 used in the communication of these modules are insufficient to meet the increased bandwidth requirements of today's aircraft. However, these legacy networks are highly reliable and deterministic, as avionics systems require. For this reason, different technologies have been developed without losing their quality of service and suitability for critical systems. Avionics Full-Duplex Switched Ethernet (AFDX) protocol, a special application of ARINC-664 Part 7, patented by Airbus, has come to the fore in recent years. Another prominent solution is the Time-Triggered Ethernet (TTEthernet) protocol, patented by TTTech. This paper compares these two protocols from the avionics perspective with a simulation model in OMNET++ and outlooks the new generation avionics networks.**

Keywords—avionics, AFDX, TTEthernet, networks, performance analysis, latency, jitter, bandwidth, OMNET++

I. INTRODUCTION

In avionic systems, it is vital that the packet delays are predictable in the communication between the units on the aircraft and that there is no packet loss. Therefore, communication between avionics units must be deterministic. However, the need for high-bandwidth protocols has increased due to the increasing number of functions in aircraft, the number of avionic units, and the increase in the size and number of packets transmitted between units.

ARINC-429 [1] protocol was developed by Aeronautical Radio Inc. for use in civil aircraft in the late 1970s. This protocol provides high service quality by using the point-to-point connection structure between the units in the aircraft. With its 100 Kbps maximum speed, it cannot cope with the increasing bandwidth demands of today's aircraft. However, it is still used in civil and military aircraft to provide high service quality and reliability.

MIL-STD-1553 [2] protocol, which was standardized by the United States Department of Defense in the 1980s, is still used in aircraft today. Since it is a redundant serial bus with a time-division multiplexing structure, it is highly specific and reliable. Nevertheless, with a maximum speed of 1 Mbps, it falls short of meeting the increasing bandwidth needs, just as ARINC-429. To overcome the limitations of ARINC-429, the Airlines Electronic Engineering Committee (AEEC) developed the ARINC-629 [3] protocol, operating at 2 Mbps. Overwhelming wiring costs on the ARINC-429 have been improved using a triple bus structure instead of a point-to-point structure. It implements the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) media access control similar to the Ethernet version IEEE 802.3 10BASE2/10BASE5. Also, utilizing ARINC-629 was very costly. Hence, despite being the fastest avionic bus of its time, this protocol was not seen as the next big thing in avionics. Instead, the idea of using Ethernet in avionic systems was in demand in those days. ARINC-629 was implemented on the Boeing 777 and completely lost its popularity with the adaptation of Ethernet to the avionics world.

AFDX [4] protocol was first introduced by Airbus in the late 1990s during the next generation A380 aircraft development. It was later standardized as ARINC-664 Part 7 in 2004 and has been widely used since then. After the A380, Airbus used the AFDX on the A350 / A400M and the Boeing on the B787 aircraft. AFDX, with its 100Mbps speed, successfully met the increasing bandwidth requirement while improving the determinism and quality of service in avionics networks with the protocol-specific structures it offers. Researches into making IEEE 802.3 Ethernet real-time, synchronized, and fault-tolerant to meet the timing requirements of different industries such as aerospace and automotive began in the early 2000s at Vienna University. The resulting protocol was published as TTEthernet and was standardized as SAE AS6802 [5] by TTTech in 2011. TTEthernet, with its 100Mbps speed and the potential to reach 10Gbps in the future, perfectly meets ever-increasing bandwidth needs while providing real-time synchronous communication, which greatly enhances the quality of service. Sikorsky used TTEthernet in its new generation helicopter S-97 Raider in recent years. In addition to all this Ethernet utilization to safety-critical applications, IEEE 802.1 Time Sensitive Networking (TSN) group works on another Ethernet solution with better synchronization and better quality of service. Aerospace TSN Profile, SAE AS6675 [6] protocol is under development nowadays, it seems promising for future aircraft developments, and it's beyond the scope of this paper.

In this paper, AFDX and TTEthernet protocols are compared and evaluated on a scenario based on a simulation model in OMNET++ [7]. Section II presents the common aspects and differences of both protocols and shows the protocol-specific approaches to determinism improvements. In Section III, performance analysis is done comparatively, and the results are discussed. Finally, section IV concludes our analysis and offers a comparative perspective on next-generation avionic networks.

II. Related Work & Background

A. Related Work

OMNET++ is an object-oriented, C++ based discrete event simulator for network simulations. INET Framework [8] and CoRE4INET [9] contain open source network models. AS6802 TTEthernet model in CoRE4INET is chosen for this study.

B. Avionics Full Duplex Switched Ethernet (AFDX)

AFDX network has two fundamental elements: End System (ES) and switch. End Systems generate and receive traffics via switches, and switches maintain the traffic flow.

AFDX defines two main features: Band Allocation Gap (BAG) for each packet in the network and Virtual Link (VL) that is physically not present but virtually function. BAG values are set for each VL individually, and AFDX guarantees that the packets are sent in the defined BAG. AFDX took the physical point-to-point structure of ARINC-429 and made it virtual while maintaining service quality of service and increasing bandwidth. VL numbers and destinations are put in standard Ethernet packets by the ESs. Switches check the VL destinations and forward them to the corresponding ports of the switch. Packets on the VLs are checked according to their BAG values and their sequence numbers. Packets that exceed the BAG values and packets with the wrong sequence number are discarded from the flow. In this way, switches regulate the traffic and minimize the possible data corruption on the VLs. The maximum allowed jitter in a VL is defined in (1).

$$Jitter_{max} \leq 40\ \mu s + \frac{\sum(20Bytes + L_{max}Bytes)\ x\ \frac{8bits}{Bytes}}{\frac{N_{BW}bits}{seconds}} \quad (1)$$

$$Jitter_{max} \leq 500\ \mu s$$

L_{max} is the maximum frame size, and N_{BW} is the bandwidth for a given VL. 40 μs is taken as the typical fixed technological latency (τ) in (1). Then the maximum latency for a given VL is

$$Latency_{max} \leq BAG + Jitter_{max} + \tau \quad (2)$$

C. Time Triggered Ethernet (TTEthernet)

TTEthernet protocol defines three different traffic classes, namely Time Triggered (TT), Rate Constrained (RC), and Best Effort (BE). TTEthernet also uses another type of packets, namely Protocol Control Frame (PCF), which is used to establish the synchronization among the devices and switches. RC traffic is almost the same as AFDX traffic, there are VL numbers and BAG values assigned, but their names are Critical Traffic ID and BAG-accounts. BE traffic is the well-known IEEE 802.3 Ethernet traffic. TTEthernet can support BE traffic simultaneously as well as TT and RC traffic. TT traffic is the most important feature that the protocol offers. Based on a global synchronization of all the devices connected to the network, TTEthernet makes it possible to relay TT messages with a Time Division Multiple Access (TDMA) structure.

TTEthernet has a similar network structure to AFDX, there are switches and ESs, but they are configured as three different parts: Synchronization master (SM), compression master (CM), and synchronization client (SC). SM sends PCF to CM, and then CM calculates the global time based on the received PCF, and the calculated global time is broadcasted as an updated PCM to SC and other devices. In this way, no device has a conflict on keeping track of global time, and the jitter is fixed as in microseconds as the standard defines. A microsecond jitter with a TDMA structure allows us to calculate the exact latency, making the network strictly deterministic, indicating that TTEthernet is a very good solution for time-critical applications.

III. Case Study & Discussion

A. Case Study

As shown in Fig. 1 below, a scenario with one switch and three End Systems (ES1, ES2, and ES3) is chosen for simplicity.

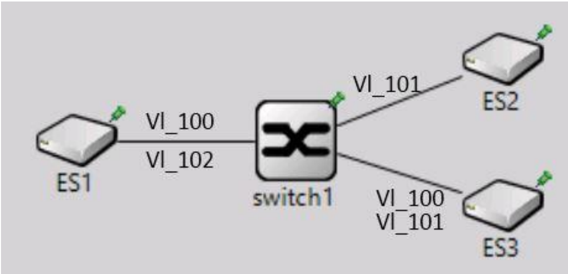

Fig. 1: Simplified Network Topology

A part of the configuration of ES2 is shown in Fig. 2, and the switch configuration with virtual link assignments is shown in Fig. 3.

```
[General]
network = small_afdx
**.ES2.phy[*].mac.address = "0A-00-00-00-00-02"
# Defines the running real-time application of
# ES2 that generates RC traffic.
**.ES2.numApps = 3
**.ES2.app[0].typename = "RCTrafficSourceApp"
**.ES2.app[0].displayName = "vl_101"
**.ES2.app[0].interval = 1ms
**.ES2.app[0].payload = 46Byte
**.ES2.app[0].ct_id = 101
# Connect the traffic generator output to the
# corresponding buffer.
**.ES2.app[0].buffers = "vl_101"
**.ES2.app[1].typename = "CTTrafficSinkApp"
**.ES2.app[1].displayName = "vl_100"
**.ES2.app[2].typename = "CTTrafficSinkApp"
**.ES2.app[2].displayName = "vl_102"
```

Fig. 2: ES2 configuration file

```
[General]
network = small_afdx

**.SW.phy[0].inControl.ct_incomings = "vl_100_ctc, vl_102_ctc"
**.SW.phy[1].inControl.ct_incomings = "vl_101_ctc"
**.SW.phy[1].shaper.tt_buffers = "vl_100, vl_102"
**.SW.phy[2].shaper.tt_buffers = "vl_100"

**.SW.vl_100_ctc.receive_window_start = sec_to_tick(995.120us)
**.SW.vl_100_ctc.receive_window_end = sec_to_tick(1015.120us)
**.SW.vl_100.destination_gates = "phy[1].TTin, phy[2].TTin"

**.SW.vl_101.destination_gates = "phy[2].RCin"
**.SW.vl_101.bag = sec_to_tick(900us)

**.SW.vl_102_ctc.receive_window_start = sec_to_tick(1020.120us)
**.SW.vl_102_ctc.receive_window_end = sec_to_tick(1040.120us)
**.SW.vl_102.destination_gates = "phy[1].TTin"
```

Fig. 3: Switch configuration file

In the first simulation scenario, we configured all three virtual links as RC to function as AFDX. BAG values and package sizes are set to 1 ms and 46 Bytes, respectively. As seen in Fig. 4, VL_101 has the maximum latency in the network as 15 µs, which complies with (2). By examining (1), one can deduce that $Jitter_{max}$ is low as expected because the frame size is chosen as little as possible in our case study. Thus the latency becomes low. More realistic scenarios can be simulated with more complex architectures, bigger frame sizes, and bigger BAG values. However, one should be careful about choosing the right values not to exceed 100Mbps maximum bandwidth.

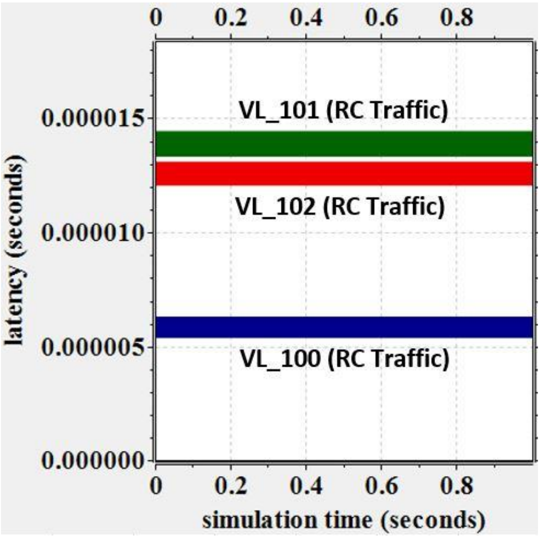

Fig. 4: AFDX simulation with using RC traffic

In the following configuration, VLs are configured as TT to see the system behavior when it operates as a time-critical TTEthernet network. As shown in Fig. 5, latency is the same for all the VLs as 35 µs for this specific configuration. It's also almost constant, just as the AFDX configuration above, which yields a high quality of service as RC traffic.

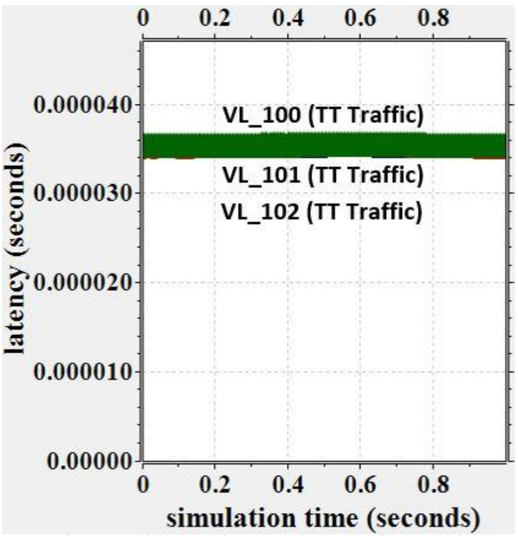

Fig. 5: TT Traffic simulation results

The next simulation result is of a mixed TTEthernet network utilizing VL_101 as RC. Others are TT. Fig. 6 shows that RC traffic and TT traffic can exist together in the same network latencies stay robust with complying with the standard. As shown, the latency of the TT VLs VL_100 and VL_102 are 35 µs and RC VL, VL_101 is 14 µs. It's important to point out that these robust results apply to this specific scenario. TTEthernet networks always prioritize TT packets. For this reason, RC traffic latency can be affected negatively if the network is not configured meticulously.

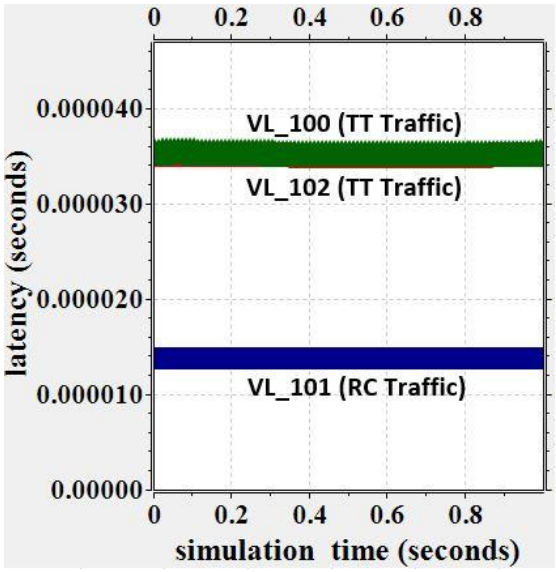

Fig. 6: Mixed traffic simulation results

Since TTEthernet networks are also capable of relaying IEEE 802.3 Ethernet packages as BE, Fig. 7 shows the configuration of ES1 to utilize BE traffic. In this configuration, VL_100 and VL_102 are TT, VL_100 is RC. ES2 and ES3, just as ES1, generate and receive BE traffic. The resulting traffic is visible in Fig. 8. Port 0, 1, and 2 of the switch show the related BE traffic.

```
[Config With_Crosstraffic]
**.ES1.numApps = 4
# Defines the running background traffic (BE)
**.ES1.app[2].typename = "BGTrafficSinkApp"
**.ES1.app[2].srcAddress = "0A-00-00-00-00-03"
**.ES1.app[3].typename = "BGTrafficSourceApp"
**.ES1.app[3].destAddress = "0A-00-00-00-00-02"
**.ES1.app[3].payload = 46Byte
**.ES1.app[3].sendInterval = uniform(200us,500us)
```

Fig. 7: BE traffic configuration in OMNET++

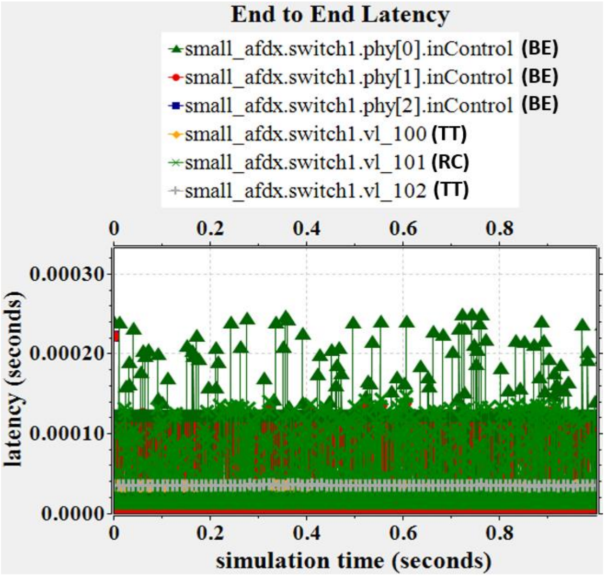

Fig. 8: Cross-Traffic simulation result

As shown in the bottom area of Fig. 8, TT traffic latencies stayed the same as 35 µs while RC traffic latency slightly and BE traffic latency greatly varies. This variation in latencies affects the quality of service negatively. One can decide to utilize BE traffic if the application is not time-critical and the quality of service is not important. Fig. 9 shows the variation of RC traffic latency smother.

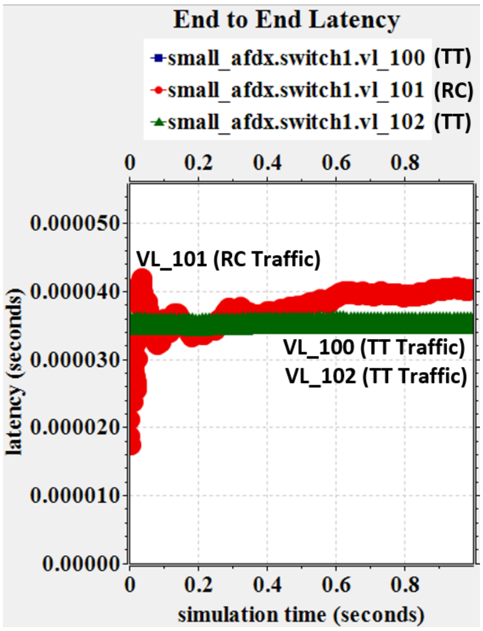

Fig. 9: Cross-Traffic simulation VL latencies

B. Discussion

Our case study shows that TT traffic in TTEthernet networks functions precisely. Using TTEthernet networks as AFDX networks by only utilizing RC traffic is also deterministic. However, adding RC traffic on top of TT traffic can affect RC traffic latency negatively while not affecting TT traffic latency. Moreover, TTEthernet architecture can utilize TT, RC, and BE traffic at the same time, but latencies of RC and BE traffics are not guaranteed to be fixed. Therefore, one must be careful about the quality of service requirements when deciding which networks to utilize and should consider the tradeoff between bandwidth availability and temporal-sensitivity. Table 1 highlights this trade-off briefly.

TABLE 1: AFDX VS TTETHERNET

Protocols / Parameters	AFDX	TTEthernet
Media Access	Full Duplex, Switched	Full Duplex, Switched, Scheduled(TT)
Bandwidth	100 Mbps (and 1 Gbps is projected)	100 Mbps, 1 Gbps and higher
Physical Layer	IEEE 802.3 physical layers	IEEE 802.3 physical layers
Bandwidth Partitioning	Statistical Multiplexing with Rate Constraints	Statistical Multiplexing for BE and RC traffic, TDMA for TT traffic
Determinism	Very Deterministic with bounded latency	Strictly Deterministic with fixed latency and µs level jitter
Redundancy	Dual Redundant	Dual, Triple or More Redundant Channels
Fault Tolerant Clock	No	Yes
Traffic Synchronization	No	Yes
Compliance with Critical Tasks	Yes	Only with TT and RC traffic

IV. CONCLUSION

AFDX protocol guarantees an upper limit of the maximum permissible jitter in the network, which consequently limits the end-to-end delay and makes it known. TTEthernet, on the other hand, is strictly deterministic as it defines a constant end-to-end delay with minimal jitter in microseconds. In this paper, we presented a simulation model of TTEthernet and we configured it as to mimic an AFDX network by only utilizing RC traffic. It's shown that TTEthernet networks can utilize more traffic and does not compromise on TT traffic quality of service. Therefore, choosing TTEthernet over AFDX or legacy protocols will be a more adequate solution to avionic platform requirements in new generation aircraft.

REFERENCES

[1] Aeronautical Radio Incorporated. ARINC specification 429P3-19, Mark 33 digital information transfer system (DITS) - Part 3 - File data transfer techniques, 2009.

[2] United States Department of Defense. MIL-STD-1553B, military standard: aircraft internal digital time division command response multiplex data bus, 1978.

[3] Y. C. Yeh, “Safety critical avionics for the 777 primary flight controls system,” 20th DASC. 20th Digital Avionics Systems Conference (Cat. No.01CH37219), 2001.

[4] Aeronautical Radio Incorporated. Aircraft Data Network. Standard 664, ARINC, Annapolis, 2002.

[5] SAE - AS-2D Time Triggered Systems and Architecture Committee. Time-Triggered Ethernet (AS 6802), 2009.

[6] SAE - AS-1A Avionic Networks Committee. Aerospace TSN Profile (AS 6675), 2019.

[7] "OMNeT++", [online] Available: http://www.omnetpp.org/.

[8] "INET Framework", [online] Available: http://inet.omnetpp.org/.

[9] "CoRE4INET Framework", [online] Available: https://core4inet.core-rg.de.