The Evolution of Aircraft Data Networks

Nour El-Din Safwat
Electronics and
Communication Department
Faculty of Engineering,
AinShams University
Cairo, Egypt

M. A. El-Dakroury
Electronics and
Communication Department
Faculty of Engineering,
AinShams University
Cairo, Egypt

Abdelhalim Zekry
Electronics and
Communication Department
Faculty of Engineering,
AinShams University
Cairo, Egypt

ABSTRACT

This paper presents the evolution of Aircraft Data Networks (ADN)'s. It describes various Avionics data network protocols based on the Avionics systems architecture. The paper describes the evolution from a simple point-to-point protocol presented in ARINC 429 to a shared data bus protocol presented in ARINC 629. Finally, the new Data network based on AFDX (Avionics Full-duplex Ethernet) is discussed. AFDX is a new standard based on Ethernet technology and able to handle today's requirements. A brief comparison between the three different protocols is presented. The comparison shows that AFDX provides better performance and flexibility without losing the compliance with safety, redundancy and reliability of Avionics requirements

General Terms

Avionics Data Buses, protocols.

Keywords

AFDX; ARINC429; ARINC629; ARINC664.

1. INTRODUCTION

Aircraft Data Networks (ADN)'s have experienced a great development since the evolution of Avionics which started as early as the 18th century. Aircraft avionics began with a few separate analog systems such as radar, navigation communication equipment and cockpit displays connected by dedicated wiring. As the aircraft became more advanced, the number of avionics per aircraft increased and systems became more complex. With the rise of digital technology, equipment was designed to communicate with each other. The use of data bus became a need in aviation. Introducing data bus protocol helped to reduce wiring which represent a considerable weight to the aircraft. It also simplifies total design and maintenance.

ARINC 429 was established as an Avionics Data bus since 1970 .it can be found in most of Civil Aircrafts. In 1980 Boeing developed a more capable digital data bus termed Digital Autonomous Terminal Access Communication (DATAC) which later became an ARINC standard as ARINC629.The increase in Avionics functionality in modern Aircraft systems leads to increase the number of electronics equipment and wiring in the aircraft. It resulted in an increase of power consumption and weight. So, changing of avionics system architectures become a need to reduce systems complexity. Figure1 shows the increase in functionality and number of electronics equipment with time.

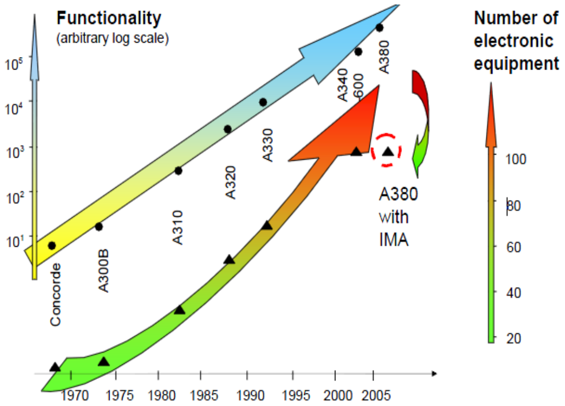


Fig1: increase in avionics systems versus time [1]

Safety and reliability are the most important Avionics requirements. Avionics data bus protocols are developed to support modern avionics systems Architectures and comply with these requirements. AFDX or (ARINC 664 part 7) is a promising Avionics network protocol. It is based on Ethernet protocol. It supports modern Avionics architectures and fulfills avionics safety and reliability requirements.

This paper is an overview on aircraft data networks. It is organized as following: First section is introduction. Second section presents the different avionics systems architectures. ARINC 429 is discussed in the third section. Forth section discusses ARINC 629. Fifth section focuses on, the new standard, AFDX and presents a brief comparison between A429, A629 and AFDX. The final section is the conclusion. .

2. AVIONICS SYSTEMS ARCHITECTURE

Avionics architectures were developed According to the order below.

2.1. Independent avionics architecture

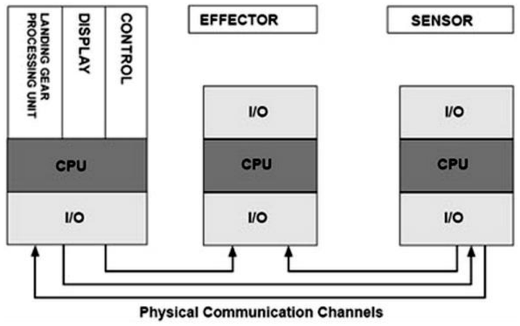
In independent avionics architecture equipment had its own functionality independent of other similar or different equipment.

2.2. Federated System Architecture

It integrates different kind of systems emerged as a Federated Avionics Systems. Separate subsystems implement functions using dedicated components, dedicated modules, LRUs, and software. Federated Architectures do not share or time-share component or information across subsystems in the avionics suite.

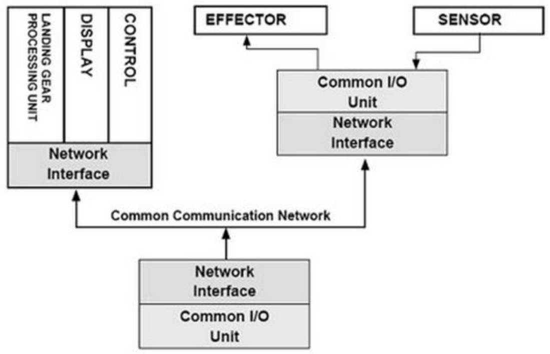
2.3. Integrated Architecture

Integrated Modular Avionics (IMA) is the new avionics architecture .it combines functions of LRU's into software packages running on a single Avionics computer. Most special-to-purpose controllers are replaced by common standardized platforms that usually host applications of several systems. IMA defines the separation of the resources and enables certification independently [2]. For inter-system communication, either IMA module-internal communication or standardized aircraft networking technology with guaranteed bandwidth and high availability is used [3]. Figure 2 and figure 3 show the Federated and Integrated architectures. It can be noticed that the number of CPUs and communication channels decreased.



Example of Federated System Architecture
CPUs: 3
I/O Modules + Network Interface Modules: 5
Physical Communication Channels: 4

Fig2: Example of Federated Architecture



Example of Integrated System Architecture
CPUs: 1
I/O Modules + Network Interface Modules: 4
Physical Communication Channels: 1

Fig3: Example of Integrated Architecture

3. ARINC 429

3.1 Network topology

In its simplest form, an ARINC 429 network consists of a single transmitter (source) connected to a single receiver (sink). Up to 20 receivers can be connected to a single transmitter as shown in Figure 4. ARINC 429 is a simplex data bus so; if any of the sink equipment needs to reply then it will require its own transmitter.

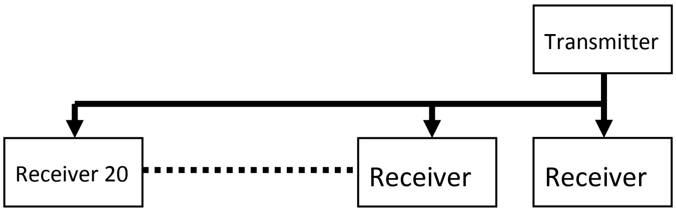


Fig4: An ARINC 429 layout with on transmitter and up to 20 receivers

3.2 Physical layer characteristics

ARINC 429 uses shielded twisted cable to transmit ARINC data in differential signal form. Data encoding is Bi-Polar Return to Zero. ARINC429 frame is 32 bits transmitted sequentially and separated by at least four bit times of NULL (zero voltage). This eliminates the need for a separate clock signal and it makes the system self-clocking.it supports two rates, low speed 12.5Kbits/sec and high speed 100Kbits/sec.

3.3 Protocol

Since there can be only one transmitter on a twisted wire pair, ARINC 429 uses a very simple, point-to-point protocol. The transmitter is continuously sending 32-bit data words or is placed in the NULL state.

3.4 Word format

ARINC frame consists of a single 32-bit data word. Figure 5 shows the five primary fields of RINC 429 (Label, SDI, Data, SSM and Parity). Bits are transmitted starting with bit 1 of the label and the final bit transmitted is the parity bit.

32	31	30	29 11	10	9	8 1
P	SSM		MSB DATA LSB	SDI		LABEL

Fig 5: ARINC 429 Frame fields

3.5 Fault tolerance features

Parity bit is the only fault tolerance feature in ARINC 429. More details about ARINC429 could be found in [4][5][6][7][8].

4. ARINC 629

4.1 Network topology

ARINC 629 is a true data bus in that the bus operates as a multiple-source, multiple-sink system as show in figure 6. That is, each terminal can transmit data to, and receive data from every other terminal on the data bus. This allows much more freedom in the exchange of data between units in the avionics system than the single-source, multiple-sink ARINC 429 topology. The ARINC 629 data bus operates at 2Mbytes/sec or twenty times that of ARINC 429.it supports up to 120 data terminal on the data bus.

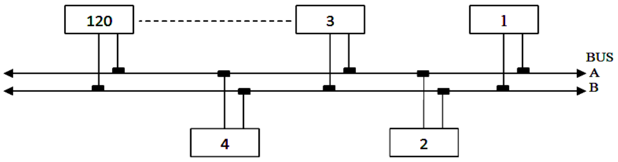


Fig 6: ARINC 629 data bus

4.2 Physical layer characteristics

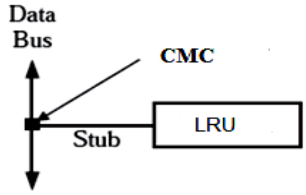


Fig7: ARINC 629 network components

The network core consists of four main components (Bus cable, CMC, Stub, LRU) as shown in figure 7.

•**Bus cable:** it is an unshielded twisted wire pair. The Bus frequency is 2Mbits/s and its Length is 100 m max.

•**Stub**: it is two shielded twisted pairs with an additional outer shield. Its length is 40 m max.

•**Current mode coupler (CMC):** The CMC provides interface between the terminal and the ARINC 629 current mode communication bus. It includes dual transmit and receive channels. The CMC is magnetically coupled to the bus by individual transformers.

•**Line replacement unit (LRU):** LRU consists of: The Serial Interface Module (SIM), Terminal Controller and Subsystem.

Subsystem generates data in form of None Return to Zero (NRZ) data .Terminal Controller converts NRZ to Manchester format data.SIM converts Manchester data to Doublet format to send to CMC then to Data Bus.

4.3 ARINC629 Protocol

There are two types of ARINC629 protocols: Basic protocol and combined protocol. Both have two modes, periodic mode and non periodic mode, but the combined protocol supports prioritized data. Basic Protocol Provides an equal priority access for each terminal to transmit either Periodic or Aperiodic data .If no bus overload exists and transmission lengths are constant, terminals transmit at a constant interval ***(Periodic Mode)*** but otherwise If the bus is overloaded, it will automatically switch to an(***Aperiodic mode)*** with no loss of data.

Because there are more than one terminal may begin to transmit when the Bus is free, a method of avoiding clashes must be used. Method used by ARINC 629 is called CSMA/CA which achieved by use of three timers:

•Transmit Interval (TI).
•Terminal Gap (TG).
•Synchronization Gap (SG).

•**The Transmit Interval (TI)** This timer is common to all terminals on the bus. It is the minimum time between two transmissions on a same terminal. TI is the longest of the three timers and starts every time a terminal begins to transmit.

•**The Synchronization Gap (SG**) This timer is also common to all terminals on the bus. The SG timer starts when the bus is quiet. It is reset if a carrier appears on the bus before it has elapsed.

•**The Terminal, Gap (TG)** This timer is unique for each terminal on the same bus. The TG timer determines which terminal transmits if two or more terminals transmit intervals has elapsed and the bus is busy. The TG timer is reset by a presence of a carrier. The TG timer should begin only after the SG timer has elapsed and only if no carrier is present.

The Basic Protocol is based on the terminals ability to satisfy three criteria's before it can transmit again:

- The TI has elapsed.
- The SG Quiet period has existed on the bus.
- The TG quiet period has existed on the bus since the SG occurred. These three rules allow transmission without collisions.

4.4 Word format

Data is transmitted in groups called Messages. Between each Message a unique Terminal Gap. Message is composed of Word-strings (31max). Word string begins with a label word followed by 0 –256 data words. There are 4 bit time gap between word-strings. Minimum word string (1 label and no data word) and Maximum (31 labels with 256 data words). Figure 8 shows an example of Message with two word strings.

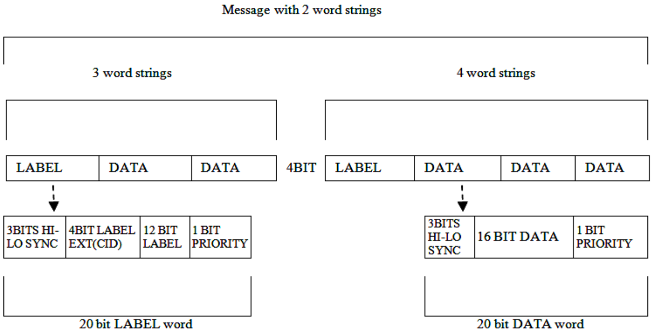


Fig8: example of Message with two word strings

4.5 Fault tolerance features

(a) Distributed control, each node has its own controller, without the need for a bus controller (avoiding single-point failure mode).

(b) Each terminal monitors its own transmissions. This is done by (SIM) which part of its functions is Fault Monitoring and Management. Checks its own output waveform, Checks received waveform, Causes/verifies coupler channel switching, Inhibits coupler transmission, if necessary and Outputs fault information to Terminal Controller.

(c) Non-intrusive, inductive coupling.

More details about ARINC629 could be found in [6], [8], [9] and [10].

5. AFDX or ARINC 664

AFDX (Avionics Full-duplex Ethernet) is a promising network protocol which comes to meet the new avionics requirements. Compared with old avionics data bus networks, AFDX supports high speed up to 100Mbit/sec, flexible architecture and low cost.

5.1 AFDX origin

AFDX is based on conventional Ethernet as it is well established protocol. Although Ethernet support high speed and low cost parts, it is based on Probabilistic protocols as CSMA/CD and non redundant architecture which don't meet the main avionics network requirements determinism and reliability. AFDX adapted the conventional Ethernet to be applicable to Avionics Network.

5.2 AFDX Network Architecture

AFDX network consists of three main parts End system (ES), switches and communication links as shown in figure 8.

•**End system** is the interface between the subsystems which transmit data and the network.

•**AFDX switch** is the central element of the AFDX network that interconnects source End System to destination End system. Links between AFDX network in twisted pair cables but they are divided to virtual links.

•**Virtual Link** defines a logical unidirectional connection from one source end-system to one or more destination end-systems as shown in figure 9 .Each Virtual Link has a dedicated maximum bandwidth. This bandwidth is allocated by the System Integrator.

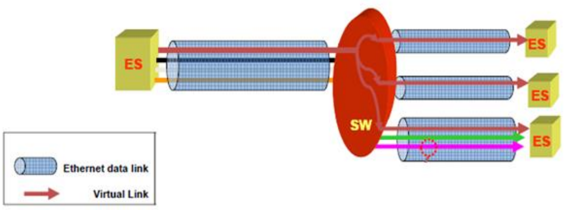


Fig 9: FDX virtual links

5.3 Physical topology

As shown in figure 10, AFDX network is a star topology network. Each ES has two AFDX ports connected to 2 redundant networks. Packets are transmitted and received over two redundant channels to ensure the reliability and availability of the AFDX standard.

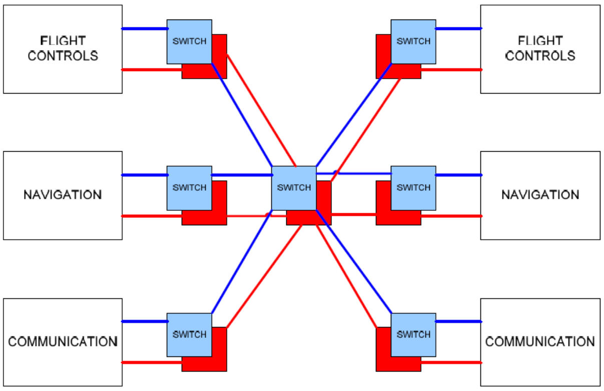


Fig10: AFDX Physical Topology

5.4 Message flow control

AFDX ensures a deterministic behavior through traffic control. Traffic control is achieved by guaranteeing the bandwidth of each logical communication channel, called a Virtual Link (VL), thereby limiting the jitter and transmit latency.

5.5 Virtual Link (VL)

The communication between two AFDX (ES) s takes place over a single physical communication link. However, it is possible to establish many logical communication links, called Virtual Links (VL). AFDX implements transmit VLs as well as receive VLs. Each transmit VL can only be assigned to one ES. Receive VLs can be assigned to several ESs.

The virtual link parameters [11]:

- ***Bandwidth Allocation Gap (BAG),*** a timeslot confining the VL's bandwidth by defining the minimum gap time between two consecutive frames. The BAG value must be in the range 1 - 128ms and must be a power of 2.
- ***Lmax***, the largest Ethernet frame, in bytes, that can be transmitted on the virtual link.
- ***Jitter*** is an upper bounded transmit latency appearing as a frame time offset within the BAG as shown in figure11.

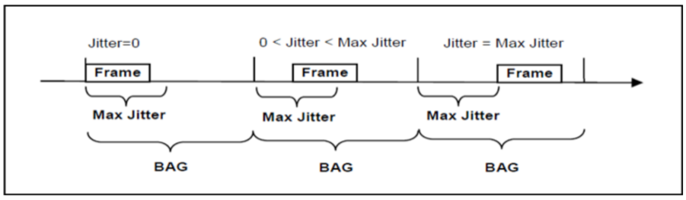


Fig11: Bandwidth Allocation Gap (BAG) & Jitter

In transmission, the maximum allowed jitter on each VL at the output of the ES should comply with both of the following formulas:

$$\begin{cases} max\_jitter \leq 40\mu s + \dfrac{\sum_{i\in\{\text{setof VLs}\}} \left(20_{\text{bytes}} + L^{\max}_{\text{bytes}}\right) \times 8\ \text{Bits/bytes}}{\text{Nbw}_{\text{bits/s}}} \\ max\_jitter \leq 500\mu s \end{cases}$$

max_jitter is in micro-seconds (*µs*); ***Nbw*** is medium bandwidth in *bits/s*; ***Lmax*** is in octets, *40µs* is a typical minimum fixed technological jitter According to the formula, the maximum allowed jitter will be lower for end-systems having few VLs and small frame sizes to process. In all cases, the jitter is bounded at 500µs to limit the impact on determinism for the whole network.

5.6 Flow Scheduling

AFDX frames are regulated based on the Bandwidth Allocation Gap (BAG) which is unique to each VL. The Virtual Link Scheduler, as shown in figure 12, is responsible for multiplexing the regulator outputs. 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).

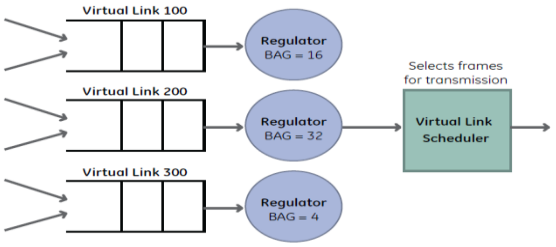


Fig12 Virtual link Scheduler

5.7 AFDX network protocol

AFDX protocol comprises five layers, as shown in Figure 13.As a protocol derived from Ethernet, The AFDX Media Access Control (MAC) data link layer is almost identical to the Ethernet MAC layer. The only difference is the so called Sequence Number (SN). The function of the SN will be explained later. On top of the Ethernet layer, AFDX implements the Internet Protocol (IP) layer which manages frame fragmentation and re-assembly as well as packet forwarding. The last protocol layer of the AFDX protocol stack is the User Datagram Protocol (UDP) which is connectionless with no transmission error control. UDP was chosen since it is more efficient than the alternative Transmission Control Protocol (TCP). Although TCP is a connection-oriented protocol providing transmission error control, this is not required since the AFDX bandwidth policing and redundancy management ensures a very low. frame loss probability. The UDP payload holds the ADFX user data [12].

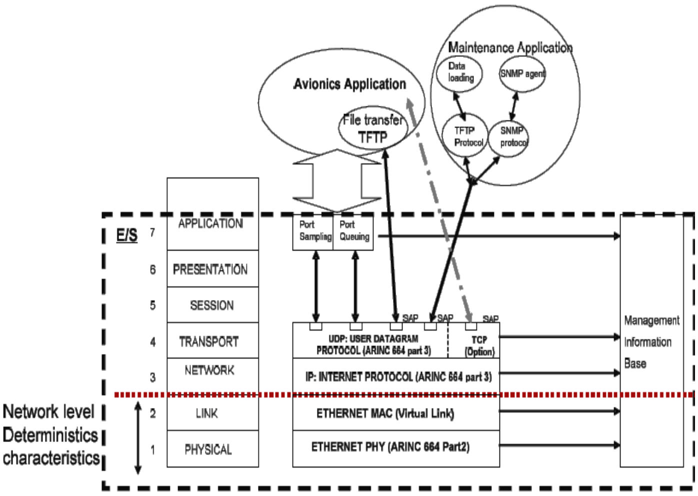


Fig13: AFDX protocol layers

5.8 AFDX frame structure

Figure 14 shows the AFDX frame structure which consists of following fields.

Preamble	SFD	MAC Destination	MAC Source	Type IPv4	IP Header	UDP Header	AFDX Payload	SN	FCS	IFG
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	20 bytes	8 bytes	17- 1471 bytes	1 byte	4 bytes	12 bytes

Fig14: the AFDX frame structure

A Virtual Link should only be identified by the MAC destination address, and the MAC source address of AFDX frames should be the MAC unicast address used to identify the physical Ethernet interface.

- **Destination address**

AFDX destination address should be compliant with the following format. As shown in figure 15 each ES should get "constant field" and "Virtual Link Identifier" values from the system integrator. The constant field should be the same for each ES in any given AFDX network. The least significant bit of the first byte indicates the group address (always = 1).In order to use the standard Ethernet frame, MAC group addresses should be used to send frames from End System to End System(s).The second to least significant bit of the first byte indicates the locally administered address (always = 1).Even if at MAC layer, only group addresses could be used, unicast communication can be seen at IP layer level by using IP unicast destination address.

- **Source Address**

The Constant field is set to "*0000 0010 0000 0000 0000 0000*" as shown in Figure 16. The least significant bit of the first byte indicates the Individual Address = 0.The second to least significant bit of the first byte indicates the locally administered address = 1.The User_Defined_ID is a single 16-bit field. It should be used as the system integrator deems appropriate to give each IP addressable host on the network a unique and meaningful IP address. The Interface_ID, indicates to which redundant AFDX network(s) the Ethernet MAC controller is connected (001 network A, 010 network B)

48 bits	
Constant field 32 bits	Virtual Link Identifier 16 bits
xxxx xx11 xxxx xxxx xxxx xxxx xxxx xxxx	

Fig15: AFDX frame destination address

Ethernet MAC Controller Identification (48-bits)			
Constant field: 24-bits	User_Defined_ID 16-bits	Interface_ID 3-bits	Constant field: 5-bits
"0000 0010 0000 0000 0000 0000 "	"nnnn nnnn nnnn nnnn"	"mmm"	"0 0000"

Fig16: AFDX frame source address

5.9 Network Reliability

The inherent AFDX features such as traffic shaping (bandwidth management and bandwidth policing) and limited transmit latency and jitter, make AFDX a deterministic and reliable network. However, in order to improve reliability even further, the AFDX network has a double, i.e. a redundant network transmitting the exact same data. The two AFDX networks are called "network A" (also "red network") and "network B" (also "blue network").

The purpose of the redundant network is to mitigate the consequences of potential network failures caused by e.g. damaged cables and connectors or devices (e.g. switches) generating babbling data.

Since ultimately only one valid data stream is required by the ES application, a handling of the redundant streams is required. As depicted in figure17 , the ES implements Integrity Checking (IC) and Redundancy Management (RM) to ensure data integrity and that only one data stream is forwarded to the upper protocol layers and from there to the application. The IC and RM services are provided by the ES without the involvement and knowledge of the application.

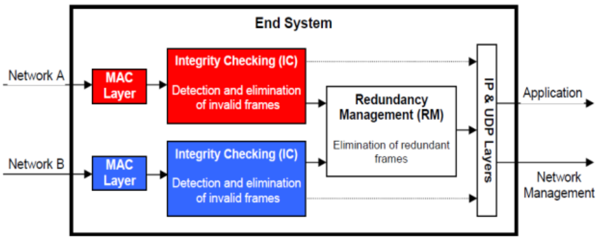

End System
Network A
Network B
MAC Layer
MAC Layer
Integrity Checking (IC)
Detection and elimination of invalid frames
Integrity Checking (IC)
Detection and elimination of invalid frames
Redundancy Management (RM)
Elimination of redundant frames
IP & UDP Layers
Application
Network Management

Fig17: AFDX Integrity Checking and Redundancy Management

- **Integrity Checking (IC)**

The first step for handling the redundant data streams is the IC, which is done separately for each network and on a per VL basis. The function of IC is to detect and eliminate invalid frames. The IC is applied on the MAC layer, i.e. on the Ethernet frame which contains a one byte Sequence Number (SN) as the last byte of the payload as illustrated in figure 4.11. The SN is the basis for the IC algorithm and is used differently in transmitting and receiving mode.

▪ **SN Usage in Transmitting Mode**

The SN is a value in the range 0 - 255 and is handled separately for each VL on each of network A and B. Prior to transmission, the SN is incremented by one for each consecutive frame (whether fragmented or not) on the same VL. With SN = 255 in the last transmitted frame, the SN is wrapped around to 1 in the following frame. Upon a reset or start-up of the transmitting ES, the SN is set to 0 in the first transmitted frame.

▪ **SN Usage in Receiving Mode**

In receiving mode, the IC uses the SN to determine if frames have been lost or whether a babbling switch is causing the same frame (with the same SN) to be transmitted over and over again. The IC algorithm accepts all frames that comply with one of the following criteria:

- SN = 0 (The transmitting ES is started or reset).
- SN = Previous SN + 1.
- SN = Previous SN + 2.

All frames not complying with these criteria are discarded.

- **Redundancy Management (RM)** [13][14]

The purpose of the Redundancy Management (RM) is to evaluate the two frame sequences delivered by the IC, discard possible duplicate frames, and forward only one copy of each frame to the upper protocol layers. The RM makes use of the configurable SkewMax parameter which is given in ms and must be specified for each receive-VL defined in the ES. SkewMax defines the maximum allowed time between receiving of the two redundant frames. The RM applies a "first-valid-wins" policy on the two frames, i.e. the first received frame is forwarded whereas the later received frame is discarded. However, if SkewMax is exceeded, the RM considers the two frames to be different from each other and hence forwards both. In the case where the RM is disabled, both frame sequences are forwarded directly from the IC to the upper layers.

5.10 Fault tolerance features

The fault tolerant mechanism which is used AFDX by is redundancy. Network redundancy overcome faults which come from physical disconnect of wires and switch failures.

5.11 AFDX versus ARINC429 and ARINC629

Table 1 summarize AFDX features and compares it with ARINC 429 and ARINC629 [15].

Table1. Comparison between ARINC429, ARINC629 and ARINC664

	ARINC429	ARINC629	AFDX
Speed	100 KHz	2 MHz	100 MHz
Throughput	2778 words/sec	Variable	Up to 3,000,000 words / sec
Latency	Fixed	Bounded	Bounded
QoS	100%	No guarantees	Configurable
Topology	point-to-point	hardware star	switched star
Reliability	the most reliable	The use of TDM guarantees delivery for reliability of function	improves the overall reliability greatly due to least hardware and wire overall
Redundancy	hardware and wire is doubled	2 busses for (standby / passive) redundancy	redundancy with dual switches
Flexibility	New subsystem is added, every subsystem it communicates with must add another ARINC 429 interface.	If another subsystem is added, it is simply coupled to the buses and hardwired for address * the latency issues	If another subsystem is added, it is simply connected to available ports on a switch

6. CONCLUSION

An overview on Aircraft data networks is presented in this paper. ARINC 429 is the most simple and reliable data bus. It widely used in the most of civil .the increase and complicity of the avionics system inside aircraft made ARINC 429 reach its limitations as it requires large number of wiring and doesn't support *IMA*.ARINC 629 which is developed by Boeing and used on B777 was a step toward *IMA*. Its cost and complexity was a reason for not be widely used. AFDX (ARINC 664) is the most promising protocol for aircraft data network as it supports the new aircraft architecture *(IMA)* and is based on the Ethernet which well established technology. Comparing AFDX with the other standard, it is the highest speed and throughput .Also; it complies with safety, redundancy and reliability of Avionics requirements.

7. REFERENCES

[1] J.B. Itier, "A380 Integrated Modular Avionics", http://www.artist-embedded.org/docs/Events/2007/IMA/Slides/ARTIST2_IMA_Itier.pdf, ARTIST2 meeting on Integrated Modular Avionics, 2007.

[2] "Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations", DO-297, August 2008.

[3] Ott, A., "System Testing in the Avionics Domain", Phd Thesis, Vorgelegt im Fachbereich 3 (Mathematik & Informatik) der Universitat Bremen,October 2007.

[4] AIRLINES ELECTRONIC ENGINEERING COMMITTEE, "ARINC SPECIFICATION 429 PART 1-17", PUBLISHED: May 17, 2004

[5] AIRLINES ELECTRONIC ENGINEERING COMMITTEE, "ARINC SPECIFICATION 429P3-18", PUBLISHED: OCTOBER 12, 2001

[6] Mike Tooley "Aircraft Digital Electronic and Computer Systems: Principles, Operation and Maintenance".

[7] "ARINC Protocol Tutorial" ZHHKTECH Co.ltd, 23 May, 2005.

[8] Janusz Zalewski, Dawid Trawczyński, Janusz Sosnowski, Andrew Kornecki, MarekŚnieŜek "SAFETY ISSUES IN AVIONICS AND AUTOMOTIVE DATABUSES".

[9] Ian Moir,Allan G Seabridge, "Civil Avionics Systems", Professional Engineering Publishing Limited, London and Bury St Edmunds, UK.,2003

[10] Alban Gabillon , Laurent Gallon ,"Availability of ARINC 629 Avionic Data Bus", IUT de Mont de Marsan, Université de Pau LIUPPA/CSySEC

[11] AFDX / ARINC 664 Tutorial (1500-049)" Condor Engineering, Inc.Santa Barbara, CA 93101, May 2005, ver: 3

[12] "AFDX® / ARINC 664 Tutorial", ID : 700008_TUT-AFDX-EN_1000, TechSAT GmbH, Poing,Release Date 29/08/2008.

[13] AIRLINES ELECTRONIC ENGINEERING COMMITTEE "ARINC SPECIFICATION 664P7", PUBLISHED: June 27, 2005

[14] Jan Täubrich, "Formal Specification and Analysis of a Redundancy Management System with TLA+", Diploma Thesis , September 21, 2006.

[15] Teresa Schuster, "NETWORKING CONCEPTS COMPARISON FOR AVIONICS ARCHITECTURE", 27th Digital Avionics Systems Conference October 26-30, 2008.