Architecting of Avionics Full Duplex Ethernet (AFDX) Aerospace Communication Network

Shabaz I Kazi
Dept. of ETE, BMIT, Solapur University, Solapur, Maharastra, India

Abstract
AFDX (Avionics Full Duplex Switched Ethernet) is an avionics data network based on commercial 10/100Mbit switched Ethernet. AFDX uses a special protocol to provide deterministic timing and redundancy management providing secure and reliable communications of critical and non-critical data. AFDX communication protocols have been derived from commercial standards (IEEE802.3 Ethernet MAC addressing, Internet Protocol IP, User Datagram UDP) to achieve the required deterministic behavior for avionics applications. End Systems (or LRU's) communicate based on Virtual Links (VL's) with Traffic Shaping by use of Bandwidth Allocation Gaps (BAGs). AFDX Switches incorporate functions for filtering and policing, switching (based on configuration tables), end-system and network monitoring.

Keywords
ADCN, MIL-1553B, ARINC 429, ARINC 664, AFDX

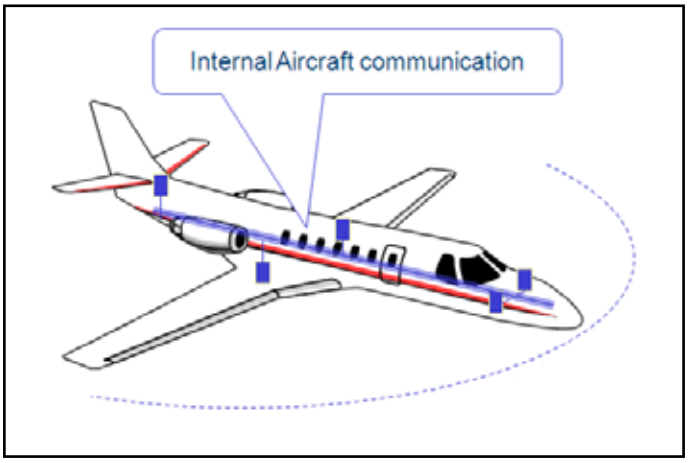

Fig. 1: General Aircraft With ADN (Aircraft Data Network)

I. Introduction

"AFDX is the standard that defines electrical and protocol specifications for exchange of data between avionics systems." AFDX standards are originally defined by Airbus in "AFDX-DFS(Detailed Functional Specification)" at the same time ARINC (Aeronautical Radio Inc.,) also has same standards defined in A664 protocol. Boeing also adopted AFDX topology in its B787-dream liner with optical fiber communication and they named it as "Interoperability specification for the 787 End systems".

The AFDX protocol is adopted in A380 (Successfully flying), Boeing 787 dream liner (Successfully flying), A400M (Yet to launch), A350 (Just finished planning) and many other. Figure 2 & Figure 3 shows AIRBUS A380 & Boeing 787 Dreamliner commercial aircrafts. Figure 4 shows the cockpit from which all the functions are controlled. The basic concepts of AFDX are derived from the Ethernet protocol and Asynchronous Transfer Mode (ATM) and following are some features of AFDX protocol

Fig. 2: Airbus A380 (AFDX Protocol)

Fig. 3: Boeing 787 Dream Liner (AFDX Protocol)

Fig. 4: Cockpit Controlled (AFDX Protocol)

High speed: 100Mbps, 1000 times faster than previous standard (ARINC429)
Less wiring: Due to the use of STAR topology for networking (on physical layer)

Less weight: Less wiring less weight. Further Use of optical cables & switches reduces the weight.
Full duplex: Simultaneous two way communication which eliminates packet collision
Robust: AFDX relies on parallel redundant network to provide reliable communication and additional level of fault tolerance.
Deterministic: Limited bandwidth, known latency & jitter, which provide required high QoS (Quality of Service)
Profiled network: All the links & their bandwidths are predefined.
Open standard: Built on OSI (Open System Interconnection) model
Practical issue: Though data rate is 100Mbps, BW is much lower than this because there should be gaps between packets to ensure that there is no Ethernet errors detected.

II. Redundancy

To increase the robustness of the system, an AFDX network consists of two redundant networks; each end system has two Ethernet ports (A and B), with A ports connected to switch A and B ports, which are connected to switch B (Fig.). Identical frames are sent by the end system on both ports simultaneously. Each switch routes their frames independently to the destination end systems. The receiving end system is responsible for managing the reception of redundant frames, deleting duplicates and any out-of-order frames.

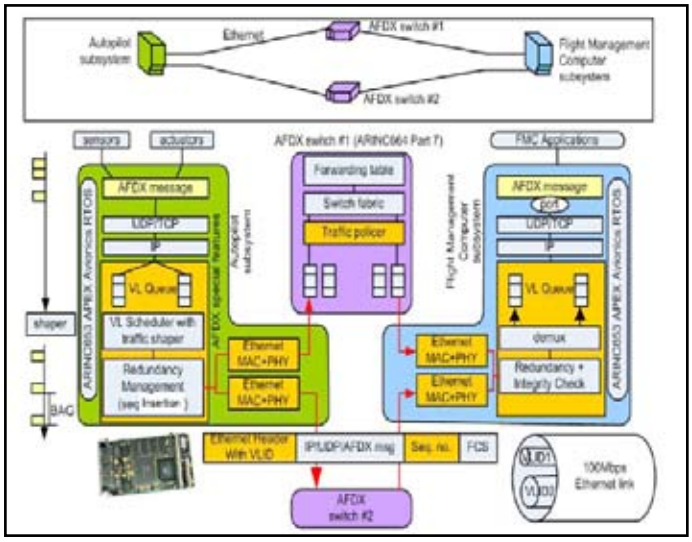

Fig. 5: Redundancy in AFDX Network

III. AFDX Message Format

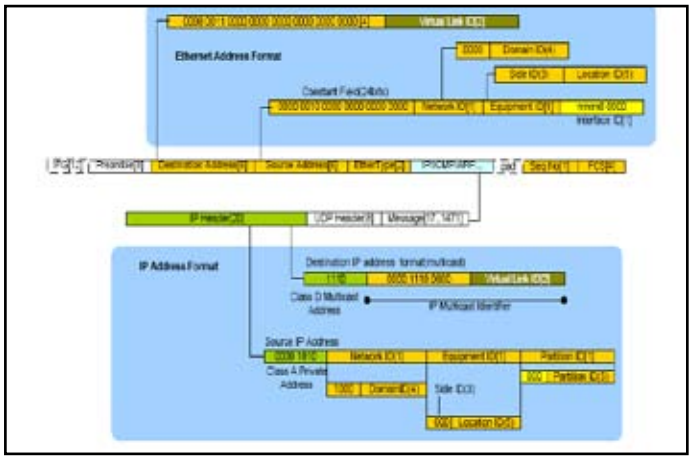

Fig. 6: AFDX Message Format

The AFDX frame format is compliant with IEEE STD 802.3 (Ethernet). The frame contains addressing for identifying source and destination end systems as well as the assigned virtual link. AFDX frame length can vary from 64 to 1518 bytes (plus a 7-byte frame preamble, 1 frame start byte, and 12-byte interframe gap (IFG), with a data payload between 17 and 1471 bytes (payload must be padded to a minimum length of 17 bytes). The one-byte frame sequence number is used to maintain ordinal integrity for frames of a given VL as well as assist in detecting missing frames. During transmission, the sequence number is incremented by one for each VL frame, starting at 0 and wrapping at 255 to 1.

VI. Arinc 429, Standard (Unidirectional, 100 Kbps/ 0.1Mbps).

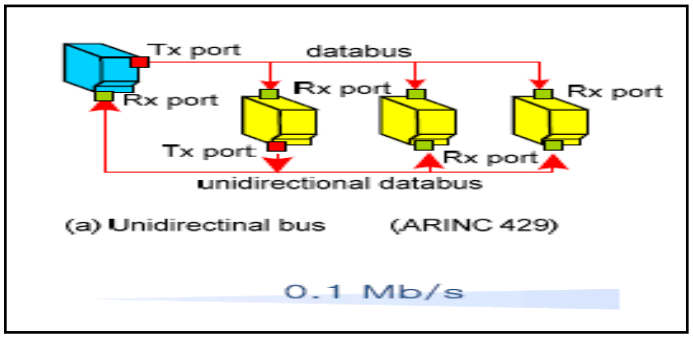

Fig. 7: ARINC 429 STD Unidirectional Bus

V. Arinc 622 Standard (Bidirectional, 2 Mbps)

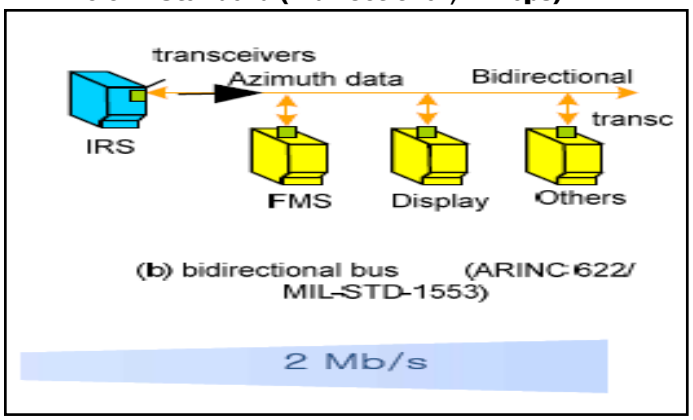

Fig. 8: MIL- STD-1553 Bidirectional Bus

VI. Arinc 664 Standard (Full duplex Ethernet, 100Mbps)

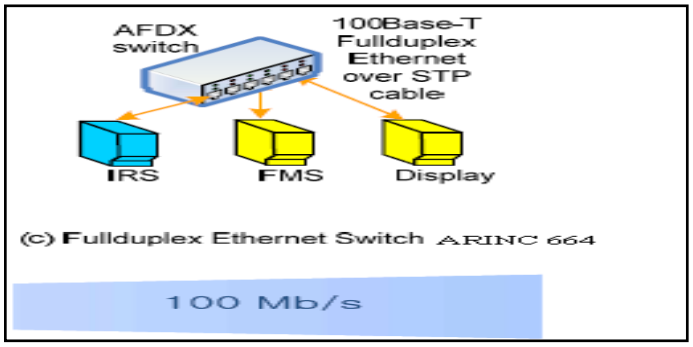

Fig. 9: Full Duplex Ethernet Network

The network is profiled — all routes and addressing are predefined and contained in the configuration for both end systems and switches, simplifying network configuration. Transmitting end systems are responsible for enforcing bandwidth limits, and receiving end systems manage redundancy. Switches are

responsible for routing frames, policing bandwidth, and shaping traffic. There is no intra-switch communication (other than passing data frames) between redundant or cascaded switches. All routes are based upon the switch's routing table. The standard (ARINC 664) also allows for the mapping of other protocols over AFDX. For example, ARINC 429 links can be built across an AFDX network through the use of concentrators/protocol conversion modules.

VII. Virtual Links

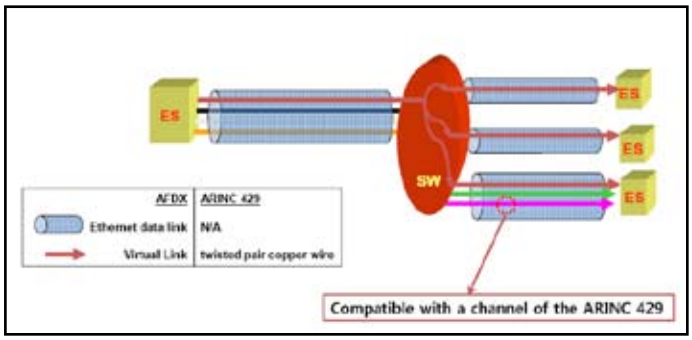

Fig. 10: Virtual Links (VLs)

An AFDX network consists of up to 24 end systems connected to a switch .Switches can be cascaded to increase the capacity of the network. Total switch capacity is limited to 4,096 Virtual Links (including the routing of Virtual Links either originating or terminating beyond end systems connected to that switch).
The goal of ARINC 664 is to preserve point-to-point links while reducing the amount of wiring. The physical point-to-point links of ARINC 429 are replaced by virtual links, connecting sensors and actuators with control units.VL links are time-division multiplexed at the end system for transmission over the network. Each VL is guaranteed a specific maximum bandwidth as well as an end-to-end maximum latency. The assigned bandwidth is controlled by the end system and enforced in the switch, where the latency is defined by the system integrator, bounded by the limits set in the standard (In addition, a VL is assigned a maximum allowed frame size of LMAX. The total of all bandwidth assigned to VLs cannot exceed the total bandwidth available in the network. Additionally, the demands on bandwidth at each switch must be known because each switch must handle VLs originating and terminating at attached end systems and any VLs being forwarded to other switches in the network. Each VL can be composed of up to four sub-VLs. Sub-VLs are used to handle less critical data with less stringent bandwidth requirements (bandwidth guarantees apply only at the VL level).Data queues for each sub-VL are read in a round-robin fashion, with each frame containing data only from one sub-VL queue (any fragmentation has to be handled at the IP layer). After a frame for a sub-VL is created, that frame is handled by the network no differently than a VL frame.

VIII. Avionics Architectures

A. Integrated Modular Avionics (IMA)

Avionics can be defined as the science and technology of electronics applied to aeronautics and astronautics or the electronic circuits and devices of an aerospace vehicle. Almost two decades ago a new concept was introduced within the avionics technology: Integrated Modular Avionics (IMA). Up to now this definition has been used for many installations onboard new aircraft types. This paper describes the basic principles of IMA and will focus on the optimal situation in which open systems concepts are combined with IMA.

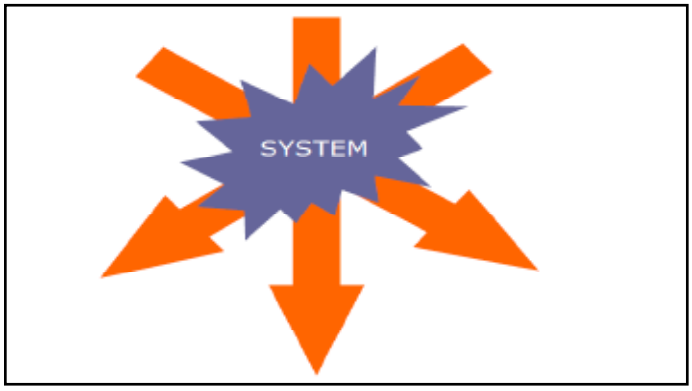

Fig. 11: Integrated System: Sharing of Data

B. Federated Style of Architectures

The traditional implementation technique for avionics can be characterized as federated, which means that each aircraft function consist of a stand-alone composition of sensors, processing units and actuators. In general data is not shared between different functions. Therefore, when looking at the avionics bay of aircraft for which the avionics suite was developed in the 20th century all contain identifiable units or "boxes" with one specific function, for example an inertial navigation system, an auto-pilot, a flight management system, a breaking and anti-skid system, etc.

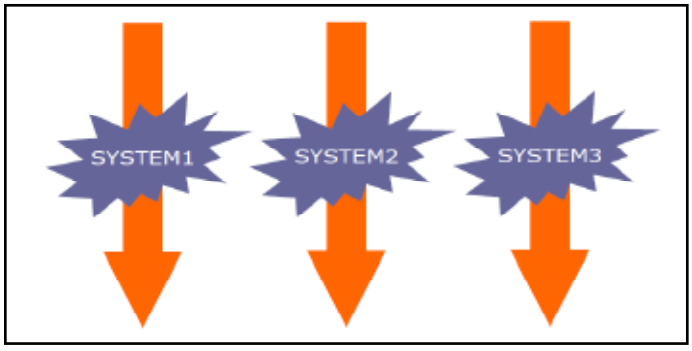

Fig. 12: Federated Architecture: no Sharing of Data

IX. Design Phase 1

Here above design shows the first attempt to simulate one to one routing via switch. Here we are trying to send a 8 bit number from one end system to the other end system via switch and it has been done successfully. The model used to test the transfer of frame from one end system to other that is from source to destination (one to one mapping).

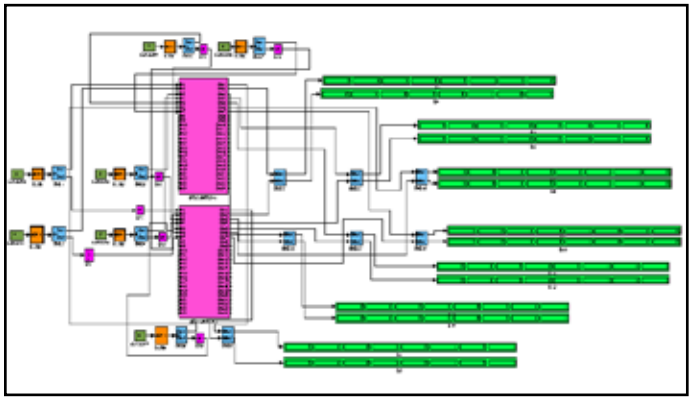

Fig. 13: ADN Network Phase 1 with One to One Mapping

X. Design phase 2

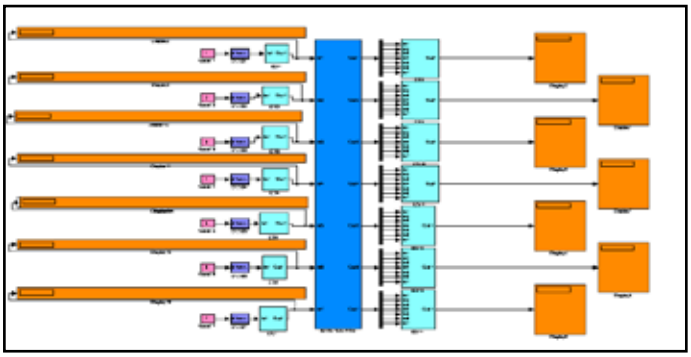

Fig. 14: Entire Design Phase 2

Above (fig.) shows the entire model of our design, which has 14 End systems, 7 End systems as Inputs (1 to 7), and 7 End systems as Outputs (8 to 14). A Frame given at the input should be shared to all the output end systems via the Switch as per the Routing Table. Example: End system 1 should send the frame to End system 8, End system 9, End system 10, End system 11, End system 12, End system 13, and End system 14. In the same way End system 1 to End system 7 should send the frame to all the output End systems that is End system 8 to end system 14.The detail explanation of each block will be explained

XI. AFDX Switch

In the above all the inputs are multiplexed and given to the embedded Matlab block in which the code is written to simulate, routing to all other end systems that is to destination End Systems. As can be seen the embedded Matlab block has 1 input and 7 outputs, that is All the inputs are shared to the other End systems with the help of Matlab code.

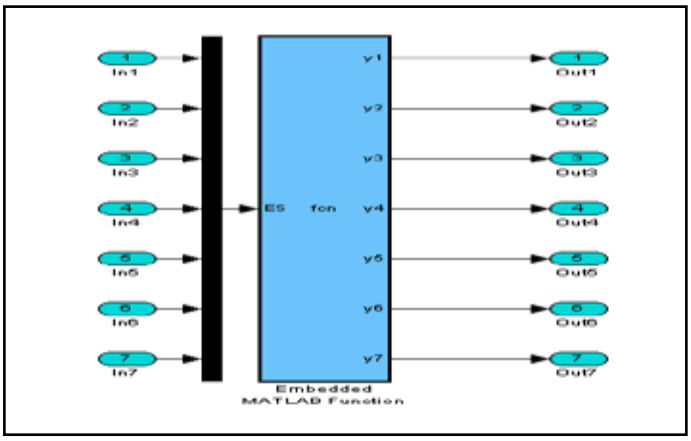

Fig. 15: Switch

At the input all the 7 inputs are multiplexed and given and at the output where all the outputs are demultiplexed and given as per routing table. The code and working of the routing table is explained in detail.

XII. Static Routing Table

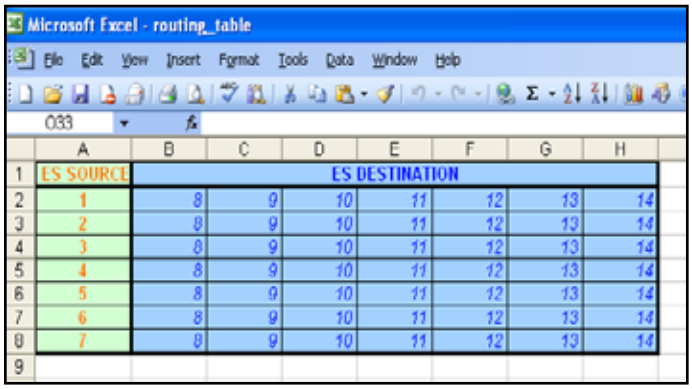

Fig. 16: Static Routing Table

Figure shows the routing table for source and destination end systems, first column shows End system source from 1 to 7,and End system destination has all 8 to 14,starting with 1 followed by the row we have 8,9,10,11,12,13,14 ,here 1 is a source and 8,9,10,11,12,13,14 are the destinations. The data from 1 should be shared to the following End systems 8, 9, 10, 11,12,13,14. And from 2 to all 8, 9, 10, 11,12,13,14 end systems. Similarly from 3, 4,5,6,7 to all other End systems 8, 9, 10, 11,12,13,14 Destinations.

XIII. Conclusion

In this project basic platform for AFDX network simulation has been prepared. The project consist of, Fourteen end systems, with seven as input end systems and seven as output End systems. We have a routing table written in excel sheet, using that we are performing the routing. All the Input values from all the Input End systems can be sent to all other Output End systems using a Embedded Matlab Code. The model shows how the routing can be done using the routing table and the Embedded Matlab code.

XIV. Scope and Future Work

AFDX Protocol can be more effectively simulated with the use of "Simevent tool set".Each port in switch are input/output port, Where as we simulated as input or output port. Should adopt the new technique for Sequence number checking for confirmation of the ordinal behavior of AFDX. Switch functions like frame filtering and policing should implemented. CRC and Hamming coding should be adopted. Switch should schedule input ports.

References

[1] "Actel Developing Solutions, Developing AFDX solutions".
[2] "A probabilistic Analysis of END–TO–END Delays on an AFDX Avionics Network", Jean-Luc Scharbarg, Frederic Ridouard, Christian Fraboul.
[3] "AFDX/ARINC 664 Protocol Tutorial", GE FANUC Embedded Systems.
[4] Ian Land, Jeff Elliott,"Architecting ARINC 664", Part 7 (AFDX) Solutions.
[5] "ARINC Tutorial", Condor Engineering.
[6] "AFDX /ARINC 664 Protocol Tutorials", TECHSAT Germany.
[7] [Online] Available: http://www.arinc.com
[8] AFDX/ARINC 664 Protocol, Tutorial – GE Fanuc

Shabaz Kazi, MTech (VLSI Design & Embedded Systems), Assistant Professor, Dept of Electronics and Telecommunication Engineering, BMIT, Solapur, Maharastra, India.