Performance Evaluation of an Ethernet-Based Cabin Network Architecture Supporting a Low-Latency Service

Fabien Geyer[1,2], Stefan Schneele[1], and Wolfgang Fischer[3]

[1] Airbus Group Innovations, Department TX4CP
Willy-Messerschmitt-Str. 1, D-81663 München, Germany
{fabien.geyer,stefan.schneele}@eads.net
[2] Technische Universität München, Institut für Informatik I-8
Boltzmannstr. 3, D-85748 Garching b. München, Germany
[3] Airbus Operations GmbH, Department Cabin Electronic Pre-Development
Lüneburger Schanze 30, D-21614 Buxtehude, Germany

Abstract. Aircraft cabin data network is a key element in today's aircraft, where several functionalities of the cabin are grouped in four different security domains. In todays architectures, each domain is normally separated from the others and uses different standards, ranging from ARINC based standards to customized Ethernet. We present here a future of cabin data network, where the main key principle is the use of a common Gigabit full-duplex Ethernet backbone, shared by all domains. As this new network has to be compliant with existing applications and their requirements, a specific Quality-of-Service (QoS) architecture is investigated in this paper. The contributions of this paper are the description of a new network architecture for cabin networks, and the introduction of a scheduling algorithm called *Time-Aware Deficit Round Robin* (TADRR) enabling an ultra low-latency time-triggered service. We show the benefits of this new architecture via a performance evaluation carried out with the simulator OMNeT++.

Keywords: Cabin Data Network, Low-Latency Service, Scheduling.

1 Introduction

During the last decades, communication networks have become increasingly present in various domains such as industry automation, automotive or avionic systems. One key technology that dominates nowadays the interest of those various domains is Ethernet, as shown in the survey from Sommer *et al.* [21]. While some Ethernet based technologies have been used for almost a decade in avionic systems such as the Avionics Full-DupleX Switched Ethernet (AFDX) technology [8,9], they are generally isolated and dedicated to one part of the complete system. This design where each function has its own system is called Federated Avionics [22]. It has advantages, such as failure isolation, but many drawbacks concerning handling and efficiency.

A. Sikora et al. (Eds.): Nets4Cars/Nets4Trains/Nets4Aircraft 2014, LNCS 8435, pp. 69–80, 2014.

We present in this paper a possible future for the network architecture used in aircraft cabins. This network is based on a shared Ethernet backbone which serves as a common access points to all cabin functions. This new backbone will offer several benefits, among them:

– less cables in the cabin which means a weight reduction for the aircraft,
– use of the well-investigated Ethernet standard which means costs reduction through the use of Commercial Off-The-Shelf (COTS) hardware components,
– introduction of true end-to-end communication in the cabin with well known protocols based on the Internet protocol suite which means the possibility to use solutions already developed for TCP/IP, and introduction of Quality-of-Service (QoS) on the network,
– ability to use standard equipment for logging and testing purpose.

In order to improve part of the end-to-end latency performances, a cut-through forwarding mechanism is used, which means that a frame is forwarded from one port to the other before the whole frame has been received. This mechanism improves the performances of the network as shown later in this paper, and with the help of a time-triggered schedule, it enables an ultra low-latency real-time service. One contribution of this paper is an architecture and a scheduling algorithm called *Time-Aware Deficit Round Robin* (TADRR) designed around this time-triggered scheduling mechanism. This algorithm is in the current trend of designing timing-aware schedulers, as proven by the recent proposal from the IEEE Enhancements for Scheduled Traffic task group [3], with the Time-Aware Shaper.

This paper is organized as followed. We first look at the related work in Section 2. In Section 3, we introduce the new network architecture which will be deployed in future aircraft cabins, the different aircraft security domains supported by this network, as well as some requirements regarding those domains. Then we present in Section 4 our scheduling architecture and algorithm proposition. We then present in Section 5 the performance evaluation of the this network, via simulations done with OMNeT++. Finally Section 6 summarizes and concludes our work, and gives an overview of future improvements.

2 Related Work

The usability of Ethernet based solutions for industrial application is a topic of interest for various industries and has long been studied. Felser presented in [12] a survey of various commercial solutions based on custom Ethernet for achieving real-time communications. A similar study on the history of Ethernet based real-time communications was made by Sauter in [18]. While the first solutions for real-time Ethernet were often based on proprietary technologies, interests have spiked recently with a work effort from the IEEE to standardize a solution for real-time communications. First dedicated to multimedia streams via the IEEE Audio/Video Bridging task group [1], this solution is now being adapted to a wider context via the IEEE Time Sensitive Network task group [4].

Regarding the avionic context, the AFDX technology [8,9] has been successfully developed as a deterministic Ethernet network. While the benefits of having a deterministic behavior are obvious, it comes at the cost of expensive custom-designed hardware. Research effort has hence been made to asses the suitability of COTS equipments for avionic systems. Performance evaluations of Gigabit COTS Ethernet switches for avionic networks were performed by Meier *et al.* in [17] as well as Jacobs *et al.* [16]. Both studies concluded that the performances in term of latency and jitter are sufficient for avionic applications. A similar study has been made recently by Suen *et al.* in [23], where they focused on the ability to complete the interchange of message between nodes in the system. They also conclude that COTS components provide performances within the range of avionic functions.

While COTS Ethernet solutions are sometimes sufficient, another affordable alternative is to use Field-Programmable Gate Array (FPGA) based solutions, as presented in this paper. As highlighted in the recent work from Carvajal *et al.* in [11], such solution can achieve better service classification and help reduce the end-to-end latency of real-time traffic.

Finally, another solution to reduce the end-to-end latency and jitter is to increase the bandwidths used in the networks. While the majority of the solutions presented in this section are based on 100 Mbps or 1 Gbps, research has also been performed regarding the usability of 10 Gbps Ethernet. With the development of COTS solutions supporting 10 Gbps Ethernet, those solutions were evaluated in the high-performance computing context, such as the work from Feng *et al.* [13], or from Bencivenni *et al.* [10].

3 Cabin Network Architecture

3.1 Topology and Node Description

The Cabin Backbone Architecture is designed in a star/chain architecture, consisting of one server in the electronic bay and up to 22 lines through the aircraft cabin. This is depicted in Figure 1. This architecture and topology is similar to the one presented in [14] and [15].

In each line up to 15 network aggregators, with functions similar to Ethernet switches, can be connected in a daisy chain manner, depending on the type of aircraft and its cabin configuration. Throughout the cabin, all devices connect to the network aggregators thus also called *Multi-Domain Network Nodes* (MD-Nodes), as they have the task to bundle the data streams from different functions and security domains onto the single backbone line while ensuring separation of traffic as well as flow control. The latter is especially challenging, as a reliable and fair scheduling for all priorities is required - independently from the position of the network node within the chain.

One specific requirement for this network is that it has to support a real-time service with ultra low-latencies. In this protocol, a packet is transmitted by the server every $31.25\mu s$, and one of the MD-Node has to answer to this packet.

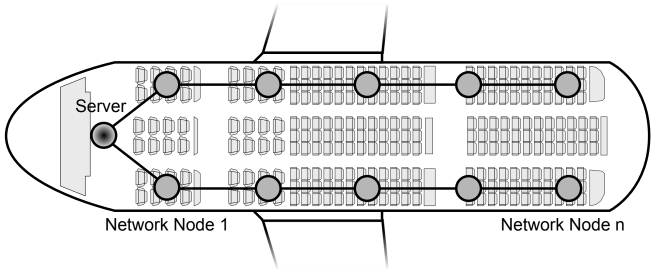

Fig. 1. Cabin Star/Chain Network Topology, here with two lines

The internal architecture of the MD-Node is presented in Figure 2. Three COTS switches are used to aggregate the traffic of each security domain. A special FPGA module is used to schedule the traffic transmitted on the common Ethernet backbone. One specificity of this FPGA is that it uses the principle of cut-through switching, which means that a frame is forwarded from one port to the other before the whole frame has been received. This is opposed to the store-and-forward principle where a frame is forwarded from one port to the other only when the whole frame has been received. A special scheduling architecture (presented in Section 4) is needed in order to prevent the traffic aggregated at this node to interfere with the traffic transmitted on the backbone.

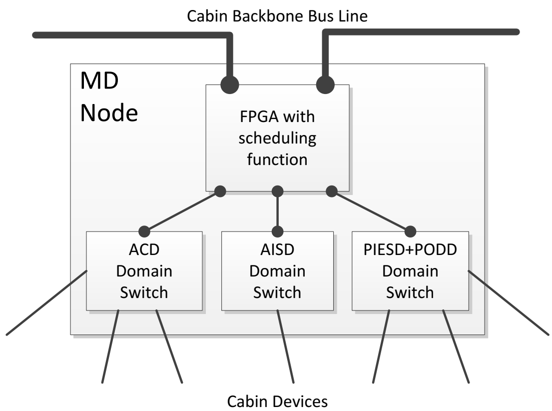

Fig. 2. Architecture of a Multi-Domain Node

3.2 Cabin Network Traffic

According to security aspects all network functions are grouped into domains. Purpose, criticality on the operation of the aircraft, and potential users are criteria for this classification. This follows the definition from [7], where domains are a superset of networks and are an accumulation of related and associated objects.

The arrangement in groups helps to handle functions with similar characteristics, because data flows in the same domain can be treated equal. The aircraft environment ARINC Specification 664 Part 5 [7] distinguishes four domains in which applications share related safety and security aspects:

- Aircraft Control Domain (ACD): all functions which are relevant to control the aircraft.
- Airline Information and Services Domain (AISD): functions to operate the aircraft and airline administrative information for the cabin and flight-crew.
- Passenger Information and Entertainment Services Domain (PIESD): functions relevant to the passenger as infotainment.
- Passenger Owned Devices Domain (PODD): passenger devices which are carried into and used within an aircraft cabin as mobile phones, or laptops. The connectivity to aircraft networks and through these to other services is provided through the PIESD [7].

Along with this traffic, we also defined a special class of traffic called Real-Time Domain (RTD), which has the purpose of scheduling the network access as well as transporting real-time information.

4 Scheduling Architecture

We describe here part of the architecture presented in [14]. This architecture is based on an addressing protocol used for scheduling the access of the MD-Nodes to the backbone, and a specially designed packet scheduler in each MD-Node.

4.1 Addressing Protocol and Real-Time Traffic

As explained earlier, due to the daisy-chain architecture and the routing requirement, the case where one MD-Node overloads a single line has to be prevented. Also, to improve the end-to-end latencies on the network, we use a cut-through mechanism. For this cut-through mechanism to be efficient, the path of a packet needs to be completely congestion free, as otherwise queuing delays may occur. Hence we need a way to avoid those congestions. The solution adopted here is to use a Time-Division Multiple Access (TDMA) architecture, where each MD-Node is allowed to send packets on the backbone only during certain time-slots. In order to avoid the use of clock synchronization protocols to distribute the time-slots, a special network protocol is used, where the server addresses each MD-Node when they are allowed to transmit packets. This protocol is also used to transmit real-time audio data which require the use of the TDMA schedule.

We use here a simple round-robin addressing, where a MD-Node is addressed every $31.25\mu s$, as presented in Figure 3. The length of $31.25\mu s$ is derived from the audio bandwidth of the current Cabin Management System. This means that each MD-Node is allocated a bandwidth of

$$\frac{B_{\text{backbone}} - B_{\text{real-time protocol}}}{\text{number of network node in the line}} \tag{1}$$

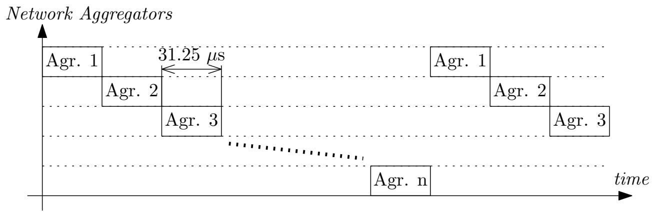

Fig. 3. TDMA with round-robin schedule

and is able to forward packets on the backbone every

$$31.25\mu s \cdot \text{number of network node in the line} \quad (2)$$

4.2 Packet Scheduling: Time-Aware Deficit Round Robin

As a MD-Node is able to forward packets only in its allocated $31.25\mu s$ time-slot, we need to schedule the packet forwarding function. We introduce for this purpose a new packet scheduling algorithm, called *Time-Aware Deficit Round-Robin* (TADRR). It is a variant of the well known Deficit Round-Robin (DRR) scheduler presented in [20]. The original DRR scheduler was designed to be a work-conserving scheduler, which means that the scheduler is idle only when there are not packet available. In our usecase, as packets are not allowed to be forwarded when the MD-Node is not allowed to send, a non work-conserving scheduler is needed. The TADRR scheduler mixes the two following functionalities:

- It is time-aware, meaning that the scheduler respects specific timing where it is allowed to forward packets or not. This is a function currently been developed by the IEEE Enhancements for Scheduled Traffic task group [3], with the so-called Time-Aware Shaper.
- It ensures a fair distribution of the available bandwidth between the different queues or flows.

We define the two following states: `WAIT_SLOT` where the scheduler has to wait for the trigger from the server to be allowed to send, and `ALLOWED_TRANSMIT` where the scheduler is allowed to forward packets. We make the following additions to the dequeuing module from [20]:

- Lines 1 to 5: The maximum allowed packet size is computed using the end of the timeslot (*endTimeslot*) and the current timestamp (t). Note that the Ethernet inter-frame gap (*IFG*) is accounted for.
- Lines 12 and 16 to 19: We use the previously calculated maximum allowed packet size and check it against the head-of-line packet.

Note that as the original algorithm from [20], the complexity of this algorithm is $\mathcal{O}(1)$.

Algorithm 1. Time-Aware Deficit Round Robin - Dequeuing

Dequeuing module:

```
if state ≠ ALLOWED_TRANSMIT then return
allowedPacketSize ← B_backbone · (endTimeslot − t)−IFG
if allowedPacketSize < minEthernetPacketSize then
    state ← WAIT_SLOT
    return
while True do
    if ActiveList is not empty then
        Remove head of ActiveList, say queue i
        DC_i ← DC_i + Q_i
        while (DC_i > 0) and (Queue_i not empty) do
            PacketSize ← Size(Head(Queue_i))
            if PacketSize > allowedPacketSize then break
            if PacketSize ≤ DC_i then
                Send(Dequeue(Queue_i))
                DC_i ← DC_i−PacketSize
                allowedPacketSize ← allowedPacketSize − PacketSize − IFG
                if allowedPacketSize < minEthernetPacketSize then
                    state ← WAIT_SLOT
                    return
            else break
        if Empty(Queue_i) then
            DC_i ← 0
        else InsertActiveList(i)
```

5 Evaluation

We describe here the evaluation of this network which was performed under OMNeT++ [6] with the INET [5] framework, which includes the required network protocols (Ethernet, IP and UDP). We follow a Monte Carlo method of simulating multiple runs, each time with a different seed and different initialization vector. We use the following assumptions for the simulation:

- All links are set to 1Gbit/s,
- The switch processing time is set to $100\mu s$,
- Queue sizes are set to 1000 packets,
- All the addressing schemes are set to static.

Flows are generated using UDP applications at a fixed bandwidth, but using a uniform distribution for the packet size. The network supports both the real-time protocol, with its $31.25\mu s$ timing, as well as additional applications.

We use the same star/chain architecture as presented in Section 3. We study three topologies presented in Table 1. We define the *uplink* direction as the direction of the packets from a node connected on a line to the central server, and *downlink* direction as the opposite direction. The utilization columns correspond to the portion of the backbone bandwidth that is used.

The results presented here compare two operation modes of the MD-Nodes:

Table 1. Studied topologies

	Configuration		Devices			Utilization (%)	
Topology	Lines	Aggregators	ACD	AISD	PIESD	Downlink	Uplink
1	3	21	18	2	2	15.4	17.4
2	3	21	20	2	2	15.6	13.9
3	4	28	7	1	29	43.7	10.5

- the *cut-through* configuration, which corresponds to the description made in Section 3, with the TDMA schedule and the TADRR packet scheduler;
- the *store-and-forward* configuration, where the FPGA acts as a traditional store-and-forward Ethernet switch, without any considerations for the real-time protocol.

5.1 End-to-End Latency - Real-Time Protocol

We define end-to-end latency as the difference between the moment that the packet has been created in the source device, and the timestamp on which the last bit of the packet is received by the receiving device.

Figure 4 presents the maximum experienced end-to-end latency of the real-time protocol described in Section 4.1. By scheduling appropriately on the cut-through configuration the time when the Ethernet frames of the real-time protocol are sent, we are able to achieve end-to-end latencies below $5\mu s$ for both directions. This is a promising result, as it enables us to have strict feedback loops.

In the store-and-forward configuration, the end-to-end latency is much worse for the real-time protocol. We see here a clear benefit of having dedicated time-slots, where the network is contention free for the real-time protocol.

5.2 End-to-End Latency - Applications

Figure 5 presents the maximum experienced end-to-end latency of the different devices in the topology.

Regarding the downlink direction, we see a definitive benefit in the cut-through configuration. Regarding the uplink direction, we see that the round-robin schedule presented in Section 4.1 is sub-optimal compared to the store-and-forward performances, with almost an order of magnitude of difference. This can be explained by the short time window of $31.25\mu s$ where a MD-Node can only transfer a small number of Ethernet frames. This means that in some cases, packets have to wait multiple round-robin cycles in the MD-Node queue before being able to be transferred.

5.3 End-to-End Jitter Measures

We used the *interarrival jitter* definition from RFC 3550 [19] for our end-to-end jitter measurement. It is defined as the measure of packet arrival time spacing

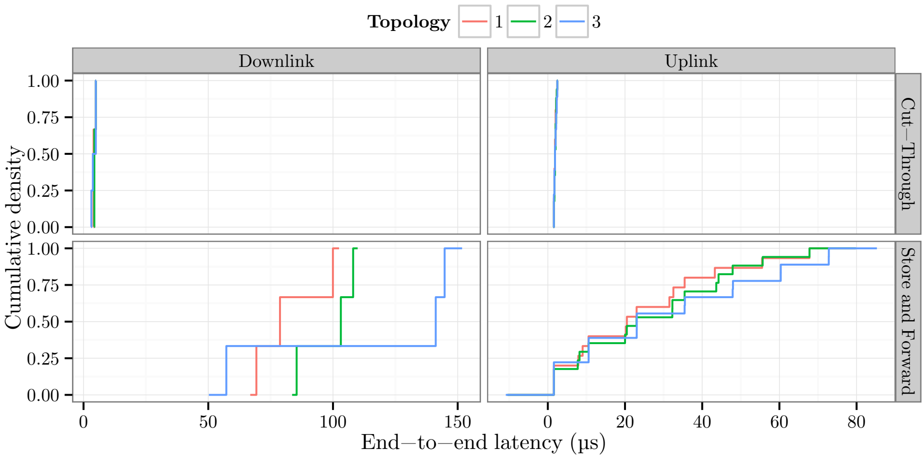

Fig. 4. Maximal end-to-end latency of the real-time protocol

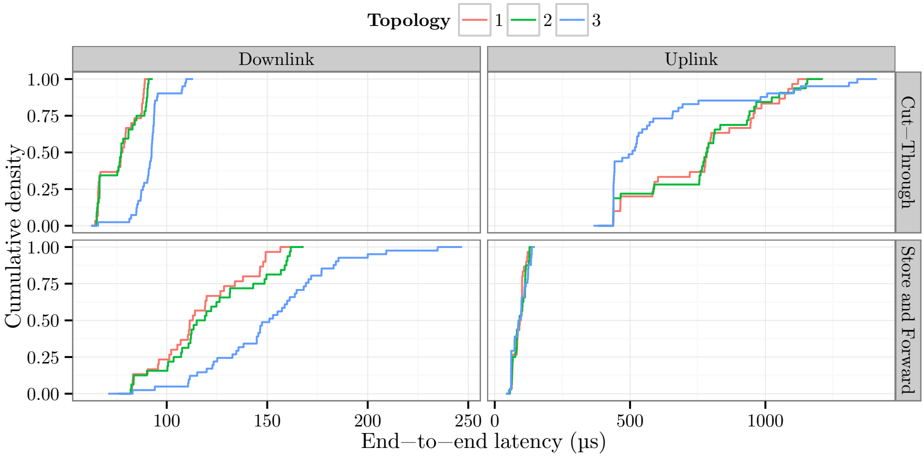

Fig. 5. Maximal end-to-end latency of the applications

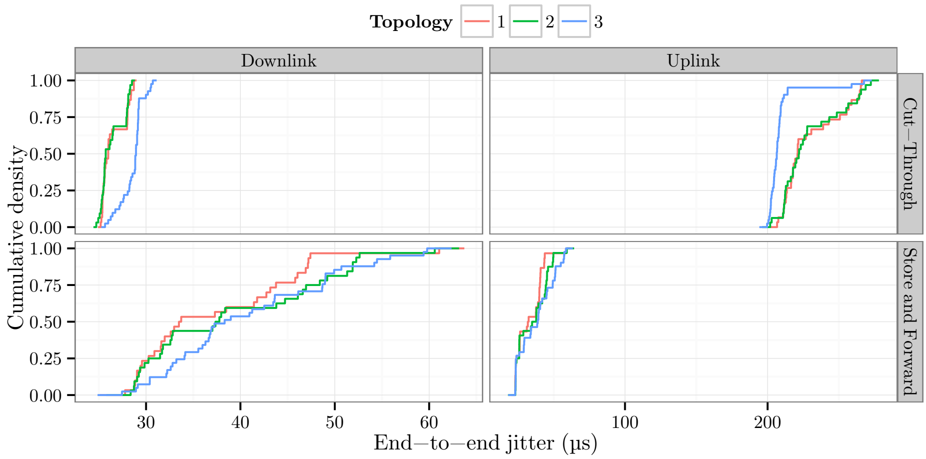

Fig. 6. Maximal end-to-end jitter of the applications

at the receiver smoothed with an exponential filter with parameter 1/16. This definition of the jitter is preferred as only non-spurious deviations in the packet spacing will affect the applications. Figure 6 presents maximum end-to-end jitter experienced by the different devices.

The remarks made for the end-to-end latency also apply for the jitter. We see here again better performances for the cut-through mode in the downlink direction, but worse ones for the uplink direction.

6 Conclusion and Future Work

We presented in this paper a new architecture for a possible future of cabin data network. The main key principle of this network is the introduction of a common Ethernet backbone, shared by the three avionic domains. As those domains share the same physical link and the network aggregators are using a cut-through forwarding scheme, some considerations have to be made regarding access to the backbone. We proposed in this paper a solution to this problem, by mixing a TDMA scheme with a new packet scheduling algorithm called *Time-Aware Deficit Round Robin.*

Via simulations performed with OMNeT++ and its framework INET, we learned some key insights regarding this architecture. While the scheduling architecture has a major benefit on the real-time traffic, and improves the end-to-end latency and jitter of downlink packets compared to the standard store-and-forward configuration, it comes at the cost of large degradations for the uplink performances. The TDMA schedule is the main cause for the good performances of the real-time protocol and the bad performances of the rest of the applications.

We would like to extend our analysis to solutions that may overcome the performance degradations of uplink packets for the non-real-time applications. Regarding the schedule of the different transmissions of the network aggregators, another algorithm than the round-robin scheme used here could bring better performances. As some network aggregators produce more bandwidth than other ones, allocating time windows in accordance to this output bandwidth seems logical. This algorithm could work offline, by careful analytical evaluation of the traffic usage at the different points of the network, or online, by extending the real-time protocol described earlier. Finally, we would like to compare the architecture presented here with new advances made by the IEEE 802.1Qbu task group [2], which is currently developing and standardizing frame preemption for Ethernet. This architecture could enable us to completely remove the TDMA schedule while keeping low-latencies for the real-time protocol.

References

1. IEEE 802.1 Audio/Video Bridging Task Group, `http://www.ieee802.org/1/pages/avbridges.html` (accessed February 25, 2014)
2. IEEE 802.1Qbu Task Group - Frame Preemption, `http://www.ieee802.org/1/pages/802.1bu.html` (accessed February 25, 2014)
3. IEEE 802.1Qbv Task Group - Enhancements for Scheduled Traffic, `http://www.ieee802.org/1/pages/802.1bv.html` (accessed February 25, 2014)
4. IEEE Time-Sensitive Networking Task Group, `http://www.ieee802.org/1/pages/tsn.html` (accessed February 25, 2014)
5. INET Framework for OMNeT++/OMNEST, `http://inet.omnetpp.org/` (accessed June 25, 2013)
6. OMNeT++ 4.3 Network Simulation Framework, `http://www.omnetpp.org` (accessed June 03, 2013)
7. Aeronautical Radio Inc.: ARINC Specification 664P5: Aircraft Data Network, Part 5 - Network Domain Characteristics and Interconnection (April 2005)
8. Aeronautical Radio Inc.: ARINC Specification 664P7: Aircraft Data Network, Part 7 - Avionics Full Duplex Switched Ethernet (AFDX) Network (June 2005)
9. Aeronautical Radio Inc.: ARINC Specification 664P7-1: Aircraft Data Network, Part 7 - Avionics Full Duplex Switched Ethernet (AFDX) Network (September 2009)
10. Bencivenni, M., Bortolotti, D., Carbone, A., Cavalli, A., Chierici, A., Dal Pra, S., De Girolamo, D., Dell'Agnello, L., Donatelli, M., Fella, A., Galli, D., Ghiselli, A., Gregori, D., Italiano, A., Kumar, R., Marconi, U., Martelli, B., Mazzucato, M., Onofri, M., Peco, G., Perazzini, S., Prosperini, A., Ricci, P., Ronchieri, E., Rosso, F., Salomoni, D., Sapunenko, V., Vagnoni, V., Veraldi, R., Vistoli, M., Zani, S.: Performance of 10 Gigabit Ethernet Using Commodity Hardware. IEEE Transactions on Nuclear Science 57(2), 630–641 (2010)
11. Carvajal, G., Wu, C., Fischmeister, S.: Evaluation of Communication Architectures for Switched Real-time Ethernet. IEEE Transactions on Computers 63(1), 218–229 (2014)
12. Felser, M.: Real-Time Ethernet - Industry Prospective. Proceedings of the IEEE 93(6), 1118–1129 (2005)

13. Feng, W., Balaji, P., Baron, C., Bhuyan, L., Panda, D.: Performance Characterization of a 10-Gigabit Ethernet TOE. In: Proceedings of the 13th IEEE Symposium on High Performance Interconnects (Hot Interconnects), pp. 58–63 (August 2005)
14. Fischer, W., Klose, P., Heinisch, M., Reuter, J.: Challenges of Future Cabin Networks. In: Proceedings of Workshop on Aircraft System Technologies, AST (April 2013)
15. Geyer, F., Schneele, S., Heinisch, M., Klose, P.: Simulation and Performance Evaluation of an Aircraft Cabin Network Node. In: Proceedings of Workshop on Aircraft System Technologies, AST (April 2013)
16. Jacobs, A., Wernicke, J., Oral, S., Gordon, B., George, A.: Experimental Characterization of QoS in Commercial Ethernet Switches for Statistically Bounded Latency in Aircraft Networks. In: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004), pp. 190–197 (November 2004)
17. Meier, J., Kim, S., George, A., Oral, S.: Gigabit COTS Ethernet Switch Evaluation for Avionics. In: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN 2002), pp. 739–740 (November 2002)
18. Sauter, T.: The Three Generations of Field-Level Networks - Evolution and Compatibility Issues. IEEE Transactions on Industrial Electronics 57(11), 3585–3595 (2010)
19. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (INTERNET STANDARD), updated by RFCs 5506, 5761, 6051, 6222, 7022 (July 2003)
20. Shreedhar, M., Varghese, G.: Efficient Fair Queuing Using Deficit Round-Robin. IEEE/ACM Transactions on Networking 4(3), 375–385 (1996)
21. Sommer, J., Gunreben, S., Feller, F., Kohn, M., Mifdaoui, A., Sass, D., Scharf, J.: Ethernet - A Survey on its Fields of Application. IEEE Communications Surveys and Tutorials (2), 263–284 (2010)
22. Spitzer, C.R. (ed.): Digital Avionics Handbook. Avionics: Development and Implementation. The Electrical Engineering Handbook Series. CRC Press (December 2006)
23. Suen, J., Kegley, R., Preston, J.: Affordable Avionic Networks with Gigabit Ethernet: Assessing the Suitability of Commercial Components for Airborne Use. In: Proceedings of IEEE Southeastcon 2013, pp. 1–6 (2013)