

Received 10 May 2022, accepted 5 June 2022, date of publication 22 June 2022, date of current version 27 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3184329

Verification and Validation Framework for AFDX Avionics Networks

JAVIER VILLEGAS¹, SERGIO FORTES¹, (Member, IEEE), VICENTE ESCAÑO², CARLOS BAENA¹, BENJAMÍN COLOMER², AND RAQUEL BARCO¹

¹CEI Andalucía TECH E.T.S. Ingeniería de Telecomunicación, Telecommunication Research Institute (TELMA), Universidad de Málaga, 29010 Málaga, Spain

²Aerospace and Defence Systems, Aertec Solutions, 29590 Málaga, Spain

Corresponding author: Sergio Fortes (sfr@ic.uma.es)

This work has been funded by AERTEC Solutions (reference 8.06/5.59.5543 and 8.06/5.59.5715) and developed in the framework of the CleanSky-2 project "Audio Suite for Disruptive Cockpit Demonstrator", reference JTI-CS2-2018-CFPO9-LPA-03-17. This work has been also funded by Junta de Andalucía and ERDF(European Regional Development Fund): project CAPTOR (advanced Avionics communications validation and verification PlaTfORM, Ref. PYC20 RE 077 UMA) and postdoctoral grant (Ref. DOC_01154 "selección de personal investigador doctor convocado mediante Resolución de 21 de mayo de 2020", PAIDI 2020. This publication has been also supported by Universidad de Málaga "II Plan Propio de Investigación, Transferencia y Divulgación Científica de la Universidad de Málaga-Ayudas para publicaciones en acceso abierto".

ABSTRACT Aircrafts fully rely in a wide variety of electronic systems: sensors, navigation equipment, representation screens and communication elements. The interconnection of these avionics implies a huge challenge for the aircraft communication networks. Extremely stringent requirements in terms of reliability and predictability have to be provided for a safe operation. There are also important limitations to recreate those networks in a prototype form, due to the costs of the related equipment and the interest of testing different architectures and possible configurations dynamically. Moreover, performance estimation approaches based solely on network calculus provide limits for worst-case scenarios, without generating a detailed feedback under different circumstances and configurations of the network and different data sources. In this context, the present work proposes and develops an evaluation framework for avionics networks in order to support the verification and validation (V&V) procedures of the communication system before applying for certification. The proposed framework implements an event-based simulator in Simulink for Avionics Full-Duplex Switched Ethernet (AFDX), one of the most extended data protocols for avionics. This is evaluated in comparison with network calculus and bibliography performance estimation approaches, demonstrating its accuracy and capabilities.

INDEX TERMS Avionics, ARINC664, AFDX, communications, verification and validation, protocols.

I. INTRODUCTION

The term *avionics* appears as the union of the words *aviation* and *electronics* and it encompasses all the electronics systems deployed in an aircraft or spacecraft. Considering the vast nature of the areas referenced by these two terms, avionics alludes to a huge variety of equipment: sensors, actuators and telecommunications systems with a wide set of uses and categories. Since the design of the A380 [1], where the concept of Integrated Modular Avionics (IMA) was introduced, the number of avionics greatly increased. The avionics are connected creating avionics networks, as summarized in the high-level diagram of FIGURE 1, these include multiple set of systems categories and applications, such as navigation, communications, monitoring,

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini¹.

flight-control, aircraft management, weather management, collision avoidance and fuel systems, among many others.

Most avionics elements require an interconnection for the communication and coordination between different systems as well as to be monitored and controlled by the crew or provide functionalities. Moreover, the resulting networks must accomplish the extremely stringent reliability needs of aircraft systems and their tight safety certification requirements [2] [3]. This has led to the definition of several standards and protocols for avionic networks which are characterized by their high level of determinism, which allows to bound delays and ensures a certain throughput.

One example of these protocols is Ethernet Audio Video Bridging (AVB) that, however, does not straightforwardly cover enough of the safety-critical requirements of avionics [4], but some of its elements are part of the Time-Sensitive Networking (TSN) standard, which it is expected

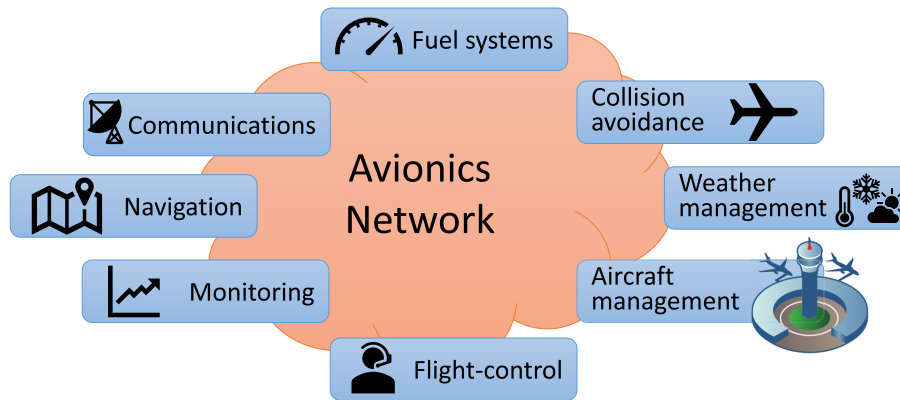


FIGURE 1. Avionics system categories.

to make its appearance in future generations of avionic networks. Other standards have been proposed, such as Time-Triggered Ethernet (TTEthernet), which is usually applied in spacecraft [5].

For modern aircraft systems Avionics Full-Duplex Switched Ethernet (AFDX) [6] is an extended solution. This protocol is an implementation of the specification ARINC 664 Part 7 [7]), which is in itself based on IEEE 802.3 [8], but with dedicated bandwidth and fixed Quality of Service (QoS). These characteristics make it suitable for engine control systems, sensors and other on-board equipment. Moreover, even though AFDX is based on Ethernet, switches must conform with the requirements and, optionally, certifications for ARINC 664, such as certifications for software (e.g., DO-178C [9] and hardware (e.g., DO-254 [10]). These, together with licensing, increases the costs of the deployment significantly.

The avionics scenario has been also impacted by the adoption of IMA [11], based on the integration of multiple related activities in the same device while guaranteeing the same security level. This, alongside AFDX providing higher capacity than its predecessors ARINC 429, allows for the introduction of more advanced and complex equipment to the aircraft.

Apart from the need of achieving the reliability requirements of avionics, the network design process should also aim to reduce the number of physical connections and therefore the weight and costs of the system. With this objective, sensors and actuators are placed near a Remote Data Concentrator (RDC) [12]. RDCs are responsible to concentrate and convert to AFDX frames the different data flows coming from peripherals, simplifying their individual interfaces.

However, the growing use of bandwidth by IMA systems pose a challenge on defining AFDX topologies compliant with the requirements of certification in a cost-effective manner. Additionally, the critical safety and certification requirements increase the time until these systems can be introduced to commercial aviation. Hence, simulations can

be a key tool to shorten the time dedicated to the development and also the verification and validation (V&V) of the requirements.

V&V processes are crucial in the aviation environment given the extraordinary requirements of reliability, availability and performance (e.g., latency, jitter) demanded. Additionally, DO-178 and DO-254 certifications are not a legal requirement towards aircraft certification, but are a good self-imposed recommendation to apply good engineering practices. In the case of AFDX, networks can be certified up to the highest grade (A) of Design Assurance Level (DAL). In this way, these certifications help to accomplish the certification requirements for airborne systems imposed by international bodies such as European Union Aviation Safety Agency (EASA) and the Federal Aviation Administration (FAA).

Furthermore, the aforementioned process is typically performed following the Model-Based System Engineering (MBSE) paradigm [13], which is important to keep traceability between the specifications and requirements of the system under development, allowing to identify emerging requirement in the earlier stages of development. For this reason, simulation tools are crucial for the steps based on Software-in-the-loop (SIL) within the process that goes from design to realization (as presented in FIGURE 2). They also can invaluablely support Hardware-in-the-loop (HIL) tests that combine partial implementations of the hardware or particular elements of it within a software-simulated environment [14]. Modeling and simulation tools are also key in validating the proposed network architecture against different scenarios, especially those representing the worst cases.

In avionics networks, the software-based simulation step is generally important for multiple reasons:

- The cost of avionics equipment, which prevents the implementation of hardware until it presents high guarantees in terms of design.
- The complexity, heterogeneity and number of elements of such networks, which limits the scope of

TABLE 1. Summary of key related works.

Category	Topic	Summary	Ref.
Optimizations	Bandwidth Reduce delay/jitter	Scheduling and routing techniques to improve network performance	[15]–[20] [21]–[25]
Worst-case analysis	Network Calculus Trajectory approach	Mathematical modelling of worst-case end-to-end delays in a network	[26]–[29] [30], [31]
Simulators	Elements Real-time Software	Details about the elements in AFDX networks simulations Tools for simulations including hardware Frameworks for software-based simulations	[32]–[34] [35], [36] [37]–[43]

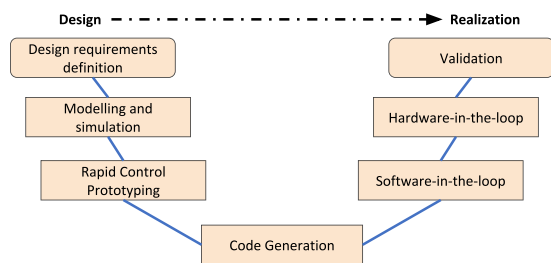


FIGURE 2. MBSE design process.

tests to be performed with hardware before the final implementation.

- The high throughput and delay requirements demanded for these networks, which imply their evaluation in multiple architectures and possible cases.
- The legal need to obtain the aircraft certification and the recommendation to get other certifications (i.e., Different Design Assurance Level depending on the risk of failure of the subsystem).
- The diversity of protocols and technologies in use and expected to be implemented in the near future, such as TSN.

Based on this, it has been deemed indispensable the creation of new approaches to support the development of avionics networks. In this field, the present work proposes a novel framework for the V&V of avionics networks via integrating network calculus and simulation modeling into a joint architecture, allowing to test new elements and software while ensuring worst-case delays. Although protocol-agnostic, in this iteration the system is focused on AFDX-based networks, also supporting the vertical integration of the designs into A653 [44] and embedded platforms such as Field-Programmable Gate Arrays (FPGAs). For such reason, the implementation is carried out Simulink where developed software can be converted to certifiable code and deployed to hardware in later stages of the certification process [45].

In this way, the present work is structured as follows. Section II analyzes the related works in the field of AFDX and avionics networks. Section III introduces the main concepts and elements of AFDX networks: end-systems (ESs), Virtual Links (VLs) and AFDX switches. Then, Section IV presents the general structure of the framework and the implementation of the simulation models. The proposed framework is

then evaluated in Section V. Moreover, Section VI discusses the main lines and open research challenges on avionics networks V&V as an outlook to future activities. Lastly, Section VII presents the main conclusions of this work.

II. RELATED WORKS

Previous works about AFDX networks can be divided in three categories as summarized in TABLE 1: optimizations, worst-case analysis and simulators.

In the first place, the optimization of AFDX networks is an important topic for the optimization of the network topology that impacts in the form of a reduction of equipment and aircraft Size Weight and Power (SWaP), which reduces costs and pollutant emissions. In this context, two lines of research are identified, one about bandwidth usage optimization and the other about reducing network delay and jitter.

On the one hand, in the topic of optimizing bandwidth allocation, the authors in [15] proposed a method for scheduling non-critical traffic. Then, the authors in [16] proposed a VL design method to minimize link usage. Later, in [17] is proposed a method to obtain optimal VLs configurations using maximum transmission units and Bandwidth Allocation Gaps (BAGs). The authors presented another optimization method for non-critical traffic using dynamic multi-path routing. Then, in [19] a hierarchical traffic shaping algorithm is proposed for Controller Area Network (CAN)-AFDX networks which minimizes bandwidth utilization. Lastly, the authors in [20] developed an algorithm to obtain optimal topologies effectively reducing minimizing weight and Operational Interruption Cost (OIC).

On the other hand, concerning network delays and jitter, the authors in [21] propose a method to enhance determinism while reducing the load by using sub-VL aggregation to avoid inserting filler frames. Moreover, in [22] is proposed a genetic algorithm that ensures the timeliness of high priority VLs and minimizes end-to-end delays in low priority traffic. Additionally, the authors in [23] present a scheduling technique for mixed-criticality traffics, which reduces the impact of non-critical traffic on the critical one. In [24] is proposed an optimal scheduling approach to reduce jitter based on frame lengths and the authors concluded that the Smallest Frame Earliest (SFE) yields the minimum average jitter. Lastly, in [25], the authors proposed a method to

determine whether break or solve the cyclic dependencies in order to obtain minimum delays and backlogs.

Nevertheless, all the works above usually need some more verification besides formal proofs, using for this purpose tools based on Network Calculus [26], trajectory approach [30] [31] or simulations. In general, most works are based on Network Calculus algorithms [26], thus results are based on a worst-case mathematical modelling. In [46] and in [47] two implementations based on Network Calculus are presented.

For this matter, most optimization works focus on adapting the arrival curves to model an AFDX network, starting by applying the serialization effect introduced by first in, first out (FIFO) queues as developed in [27] and in [28]. Furthermore, the authors in [29] apply additional constraints to Network Calculus, such as Group Strategy and their Rate-Constrained Group Strategy improving the analysis. Although oriented towards optimizations, these algorithms are limited to worst-cases, which can also be obtained through simulations that can include other characteristics as well.

Regarding simulators, the authors in [32] presented a generic simulation architecture for AFDX network, while describing the most important concepts in AFDX. Additionally, in [33] the modelling and simulation of AFDX networks is described in detail, giving an in-depth insight of the modelling of the network elements. Furthermore, the authors in [34] described the element network models to be used in NS2. Moreover, the work in [35] proposes a real-time simulator for the dSpace prototyping platform developed using Simulink. Lastly, the authors in [36] described the implementation of a simulator for an ARINC-Based hardware platform. However, these approaches require a hardware platform for testing which, in many cases, might not be available and would suppose an additional cost.

In this way, there are several works based on adapting pre-existing tools. QNAP2 is proposed in [37] as a simulation approach to bound end-to-end delays. In [42], the authors proposed a modular simulator for switched Ethernet networks called SEtSim and developed using Simulink. Furthermore, an AFDX implementation based on the TTEthernet [5] model on OMNeT++ is described in [38] and another implementation in OPNET is presented in [39] and [40]. Likewise, the authors in [41] did a study of the results obtain of simulating AFDX networks using OPNET. Finally, TrueTime [43] is a Simulink-based simulator for real-time control systems occasionally used to simulate AFDX network [23].

Generally, these previous tools are time-driven, leading to high computational costs as they require to process all the possible time steps during the simulation time. However, given the discrete nature of packet switching networks, in [48] is defined and developed a purely event-driven simulator.

III. PRINCIPLES OF AFDX

The AFDX protocol, initially proposed by Airbus [49], was designed for on-board interfaces in aerospace vehicles,

including motors, flight controls, navigation systems, as well as other critical systems for the correct functioning of the control platform. In comparison to its predecessor ARINC 429 [50], AFDX provides logical abstraction of interfaces allowing modularity, while also providing an increase in bandwidth needed for the throughput requirements of avionics. In addition to the improvement against its predecessor, AFDX is especially attractive since it is based on such a mature and widely available technology as is Ethernet, which makes its implementation easier at reduced costs.

AFDX is characterized for being a packet switching protocol, where frames are forwarded using unidirectional logical paths called *Virtual Links (VLs)* from a particular source system to the destination or multiple destinations. Network traffic is properly shaped to be transmitted during time intervals dedicated to each VL, which are called BAGs. Moreover, each VL is statically routed and previously defined to operation, which alongside with frame sizes and BAG known beforehand provide determinism to the network. In this way, AFDX is a rate constrained Ethernet, hence, each VL has a guaranteed transmission bandwidth and the jitter and delay are bounded if there is enough bandwidth for all the VL. In this way, the jitter at the output of the ES should comply with the two limits established by Equation (1) [7]:

$$\text{jitter}_{\max} \leq \begin{cases} 40\mu\text{s} + \frac{\sum_{i=1}^N (20 + L_{\max}) \text{ bytes} \cdot 8 \text{ bits/byte}}{R \text{ bits/s}} \\ 500\mu\text{s} \end{cases} \quad (1)$$

where L_{\max} is the maximum packet size for each VL and N is the number of VLs of the ESs, R represents the mean bitrate available in the physical link, $40 \mu\text{s}$ is a typical minimum fixed technological jitter and $500 \mu\text{s}$ is an upper bound to limit the impact on determinism for the whole network.

In relation to the network topology, this is formed mainly by two types of elements, the ESs and the *switches*, which will be described in the next subsections.

A. END-SYSTEM (ES)

The network entities communicating the IMA equipment in an AFDX system are the ESs. Each one acts as an interface between the IMA equipment and the AFDX network, receiving data from one or several systems and sending it through the network.

FIGURE 4 shows the logical functional blocks of the ESs. Data flows are multiplexed using VLs, being able to have one or more ES as recipients, although messages are addressed to a particular VL, and being all the routing for each VL statically defined. Additionally, there is usually VL redundancy, which means the data of a VL is sent through two different paths simultaneously, typically using two identical networks. The recipient is in charge of managing the redundant packets, usually applying a *First-Valid-Wins* [51] policy. From a physical link perspective, the packets are duplicated by the *redundancy management* block

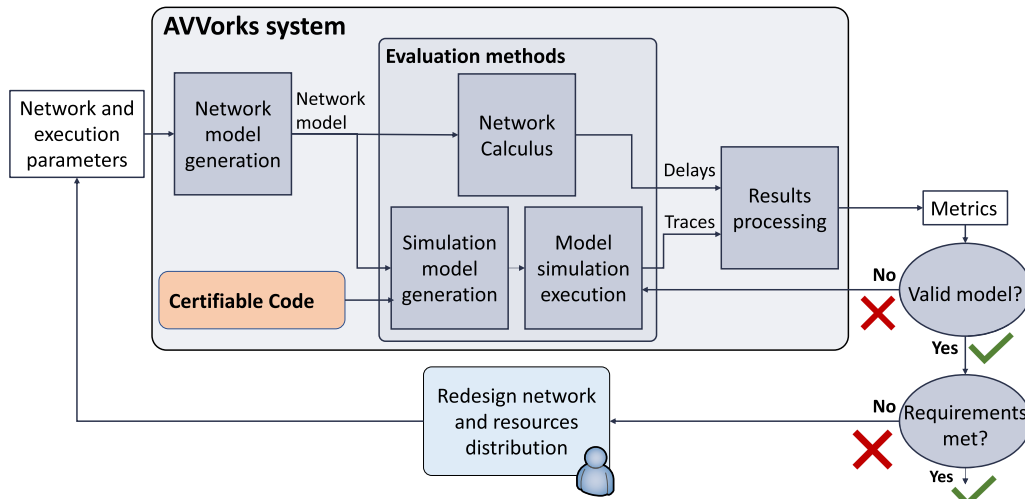


FIGURE 3. Architecture of the proposed V&V framework.

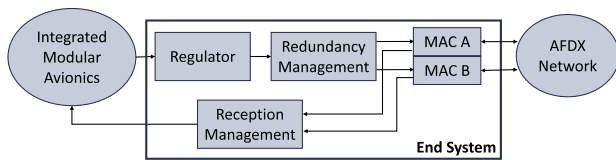


FIGURE 4. AFDX end-system logical architecture.

and sent through two redundant interfaces using two message authentication code (MAC) interfaces, generically named as MAC A and MAC B.

The ES is also responsible of shaping and multiplexing network traffic using the *regulator* block in order to minimize the jitter. In the receiving end, the ES receives packets from both networks and manages the redundancy in the *reception management* block.

The regulator block is also responsible to control what the size of a packet is and when it is transmitted, by buffering the incoming data from avionics and software partitions and transmitting the data in a deterministic way according to the corresponding VL configuration for each data flow. In this way, each data flow is guaranteed to have a dedicated bandwidth and network jitter is bounded by constraining the data rate of each VL and periodically transmitting frames according to the BAG, as it is shown in FIGURE 5. As depicted, only one packet per VL can be sent in a BAG. Additionally, the elapsed time between the start of the BAG and the start of the transmission must be lower than the maximum jitter allowed as presented Equation (1).

B. SWITCH

The other key elements in the AFDX network are the switches, which are responsible of delivering the packets to their destination. The routing is performed at Data-Link level,

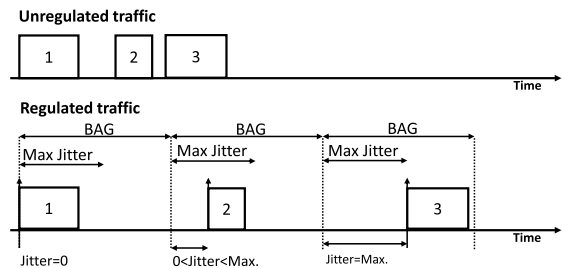


FIGURE 5. Traffic shaping in an AFDX end-system.

therefore, layer 2 switches are used for this purpose utilizing the VL identifier in the frame header, with up to 4096 values.

The switches are also in charge of applying traffic policies to enhance the determinism in the network. In this way, the application of scheduling and filtering policies is positioned as the key for narrowing down jitter and delays in AFDX networks. Here, Round-Robin [52] and Token Bucket [53] are traditionally used as scheduling and filtering policies, respectively. The filtering policy ensures the determinism in the network by discarding those packets not configure of exceeding the BAG.

IV. PROPOSED FRAMEWORK

As indicated in Section I, obtaining the necessary metrics to assess the performance of avionics networks is a crucial task for their development, verification and validation. Typically, the requirements to be validated and verified in the network is whether the delay bounds for the avionics to work properly for each VL are met or not. With this objective, a novel Avionic V&V networks Framework (AVVorks) is defined.

The proposed AVVorks architecture is shown in FIGURE 3. As inputs for the framework there are the network and execution parameters defining the system to be evaluated. These are summarized in TABLE 2. Execution

TABLE 2. Input configuration parameters.

Parameter	Fields
Execution Time	Duration in seconds
BER	Bit Error Rate
Throughput	Transmission speed in bits/s
Topology	Identifier ESs Route A Route B BAG (ms) Max packet length (bytes) Min Packet Length (bytes)

time, Bit Error Rate (BER) and throughput are directly specified. The BER is the number of errors per bit transmitted and, therefore, quantifies the quality of a transmission channel parameter. This is configurable for each of the element in the simulation, establishing the bit error rate for a connection. However, this parameter is not relevant in topologies with two identical redundant networks and, for this reason, is not evaluated in the simulations performed in this work. Topology details are established by defining the characteristics of each VL: each pair of interconnected ESs, the two routes between them (from origin to end) and the configuration of each link in terms of BAG, maximum and minimum packet length and VL identifier.

From these inputs, the *network model generation* uses the information from the inputs to generate the network model with all the specified ESs and switches. Moreover, the model associates each VL to the correspondent ES and establishes all the connections between ESs and switches.

For the generation of the model, the network topology is described as a list of VLs containing the source ES, routes and link configurations. This is then used to obtain the different ESs and switches in the network, as well as to get the routing tables of each element. With these tables, the model is set up using them in block masks to configure the different elements and connections. From the routing tables and connections, it can be verified if the paths are established correctly. Moreover, the framework allows to set data scopes and view the entities which went through the selected interfaces during the simulation if further verification is deemed necessary.

The generated network model is then executed by the *evaluation methods* including both simulation and Network Calculus approaches. In the case of Network Calculus, these results are the worst-case end-to-end delays. Meanwhile, for the simulations, the results are in the form of traces containing the logging of the packets transmitted and received by each element of the network. From this data, packet delay and jitter can be calculated, as well as throughput and the number of packets discarded by errors or by the filtering policy. The generated results are then processed to generate the output metrics, typically throughput, jitter and delay. The throughput is calculated as the volume of data transmitted in a packet

divided by the BAG, the delay as the difference between the reception and transmission timestamps and the jitter is the delay minus the mean delay of the VL.

These metrics can be used to assess whether the network is properly modelled or not and check if the requirements are met. For the latter, throughput metrics can be used to determine the link usage and, therefore, the determinism of the network, while the delay and jitter metrics would allow to assess if the temporal requirements are met. For the former, to evaluate the quality of the simulation model, its results can be compared with the Network Calculus outputs. This is done as other state-of-the-art frameworks are focused on worst-case end-to-end delays, therefore, the validation of the proposed models is based on these delays. For the validation, two main mechanisms are adopted, so the results are assessed from the perspective of two different techniques. The first one is the Exact Possible Longest (EPL) method, which computes worst-case delays using the Model Checking approach [54], which is an approach based on Timed Automatas. As a shortcoming, this method is not suitable for large amounts of VLs because of the combinatorial explosion problem. The second method is the Burst kept Network Calculus with Optimized Grouping Strategy (BNCOG) [29] which is an optimized Network Calculus algorithm.

Once checked the correctness of the modelling, the resulting metrics are applied to identify if the proposed network meets the established requirements and to study its behavior. If the modelling is correct and the network suits the specifications, the development can move to the next step towards certification. However, if the proposed network does not pass the test, the network has to be redesigned and/or its resources redistributed, and a new evaluation must be carried out.

As defined, the AVVorks framework focuses on modelling the Data Link Layer packet transmission. For this reason, the simulation is performed by the definition of the interfaces and the generation and management of packet entities of the elements in the simulation model (e.g., ESs and switches). Therefore, the key to achieve reliable evaluation tools lies in the models used for the ESs and the switches, as described in the next subsections.

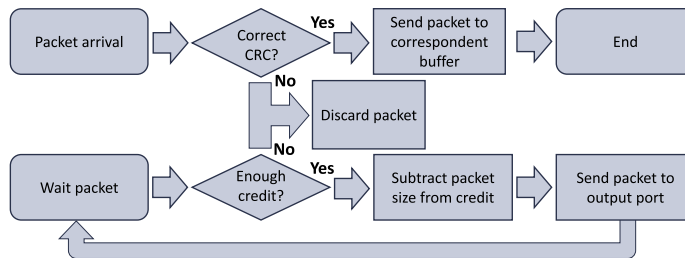
A. END-SYSTEM MODEL

The ES model has two parts, the receiver and the transmitter. The receiver contains a FIFO for each input port and combines the timestamps of the packets once it is correctly received. In the transmitter side, on-board equipment is emulated through a packet generator source, redundancy management and the route selector module.

The ES process follows the logical scheme shown in FIGURE 6a. A timer is set when packets are generated, which are then duplicated by the redundancy management and sent through the corresponding physical interfaces by the route selector. Queuing, switching and transmission times of each given packet are modelled. Lastly, in order to model transmission errors, the Cyclic Redundancy Check (CRC) is



(a) Logic followed by the ES model.



(b) Flowchart of the switch functionality.

FIGURE 6. High-level schemes of the AFDX elements functionality.

set before sending the packet according to the established BER of the link.

This delay modelling can be expressed by the equation below:

$$\begin{aligned}
 \text{Delay} &= d_{tx} + \sum_{i=1}^N (d_{tx_i} + d_{switch_i} + d_{queue_i}) \\
 &= (N + 1)d_{tx} + N \cdot d_{switch} + \sum_{i=1}^N d_{queue_i} \quad (2)
 \end{aligned}$$

where d_{tx} is the transmission time of a packet, d_{switch} the delay introduced by the switching, which can be considered common for all switches and d_{queue_i} is the queuing delay of each of the N switches in the path. Hence, d_{tx} and d_{switch} are added as constants to the delay on each of the blocks, while the d_{queue} depends on the functioning of the switch.

Therefore, from the perspective of the simulation the ES is responsible of generating the packet entities for each VL at the right time and with the appropriate characteristics according to the VL parameters and applying the first delay corresponding to the transmission.

Considering this, the worst-case is established by setting the transmission start time of each ES, in order to make VLs packages to arrive to certain switches at the same time and maximized the queue time of the VL whose worst-case is being calculated.

B. SWITCH MODEL

The switch model consists of a FIFO queue for each input port, the scheduler and output queues for each interface. Its functioning is based in two processes, the scheduling algorithm and the filtering policy, respectively. These processes are shown in a flowchart in FIGURE 6b. The first one corresponds to the upper part of the figure, which checks the integrity of the packets and selects them following the Round Robin algorithm. If the packet does not have a

correct CRC, this means, errors are detected in it, is discarded. If not, it is sent to a buffer where the filtering policy is applied. This is presented in the lower part of FIGURE 6b. In this case, the filtering policy applied is Token Bucket which, for every packet received, and if enough credit is available the packet is sent, otherwise it is discarded.

Regarding the simulation, the switch receives each packet and places it in a queue, then selects the packet to be transmitted from the queue according to the scheduling algorithm, applies the policy filtering. Finally, it applies the technological and transmission delays to the packet before sending it through the corresponding output port.

V. EVALUATION

This section presents the details on the real-world implementation of the proposed framework and the assessment of its capabilities to evaluate avionics networks. For this, two key examples use cases will be analyzed.

Here, the AVVorks framework has been implemented using Matlab and Simulink. Given the discrete nature of packet switching networks, a purely event-driven simulator [48] approach is adopted. In this type of simulation only those time instants with an event are evaluated, as opposed to the bibliography time-driven simulators which requires sampling the along the simulation time. This makes event-driven simulations generally faster and more suitable for switching networks simulations. This type of simulation is used in other schemes, such as the validation of Hardware Description Language (HDL) code in the design of FPGAs and Application-Specific Integrated Circuits (ASICs).

In this way, the network model is used to generate a Simulink environment with all the networks elements configured with their simulation parameters. At this point, certifiable code can be generated so it can be used in the following stages of the project life cycle, as a system constituted by individually certifiable elements is more prone to be certifiable as well. This is usually done using tools such as SimulinkCheck and Polyspace [55].

TABLE 3. Network setup for the use case A.

Transmitter	Virtual Link	Receiver	Route	Packet Size	BAG
ES1	V1	ES6	$ES1 \rightarrow S1 \rightarrow S3 \rightarrow ES6$	500 B	4 ms
ES2	V2	ES7	$E2 \rightarrow S1 \rightarrow S3 \rightarrow ES7$	500 B	4 ms
ES3	V3	ES6	$ES3 \rightarrow S2 \rightarrow S3 \rightarrow ES6$	500 B	4 ms
ES4	V4	ES6	$ES4 \rightarrow S2 \rightarrow S3 \rightarrow ES6$	500 B	4 ms
ES5	V5	ES6	$ES5 \rightarrow S3 \rightarrow ES6$	500 B	4 ms

TABLE 4. Worst-case end-to-end delays for the use case A.

Virtual Link	Transmission start time					Evaluation method		
	ES1	ES2	ES3	ES4	ES5	EPL	BNCOG	Simulation
V1	$2\Delta t \mu s$	$\Delta t \mu s$	$0 \mu s$	$0 \mu s$	$96 \mu s$	$272 \mu s$	$272 \mu s$	$272 \mu s$
V2	$0 \mu s$	$\Delta t \mu s$	$0 \mu s$	$0 \mu s$	$96 \mu s$	$192 \mu s$	$192 \mu s$	$192 \mu s$
V3	$\Delta t \mu s$	$0 \mu s$	$2\Delta t \mu s$	$\Delta t \mu s$	$96 \mu s$	$272 \mu s$	$272 \mu s$	$272 \mu s$
V4	$\Delta t \mu s$	$0 \mu s$	$\Delta t \mu s$	$2\Delta t \mu s$	$96 \mu s$	$176 \mu s$	$176 \mu s$	$176 \mu s$
V5	$\Delta t \mu s$	$0 \mu s$	$0 \mu s$	$0 \mu s$	$96 + 2\Delta t \mu s$	$272 \mu s$	$272 \mu s$	$272 \mu s$

Results for EPL (Exact Possible Longest) and BNCOG (Burst kept Network Calculus with Optimized Grouping Strategy) extracted from [29].

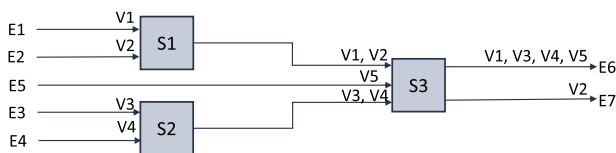


FIGURE 7. Avionics network topology for use case A.

A. NETWORK USE CASE A

The example AFDX network use case considered in [29] has been implemented in order to demonstrate the correctness of the AVorks framework and its usability in the evaluation of AFDX network topologies. The use case consists in the computation of the worst-case delays that could occur applying different constrains to the Network Calculus algorithm reducing its pessimism. For this matter, an avionics network topology defined in [29], with five transmitters and two receivers, is depicted in FIGURE 7 and characterized by the configuration presented in TABLE 3.

For this case, worst-case scenarios are summed up in TABLE 4, where Δt is a negligible time used to set the order of the packets in the queues. This means $\Delta t \ll 1 \mu s$ is several orders of magnitude smaller than the delays and it does not affect the calculation of end-to-end delays. The table shows the time when each ES has to start transmitting to obtain the maximum delay for each VL and the worst-case end-to-end delays for three evaluation methods.

In AFDX networks the worst-case scenario for a given VL occurs when the backlog of the switches in the path of the VL is maximized. The backlog is defined as the number of bytes waiting in the queue to be transmitted. In this way, the maximum backlog is given when a switch receives at its input ports at the same time the longest possible packets from the VLs competing for a common output port.

Moreover, the TABLE 4 shows the delays obtained from the three implemented methods: EPL, BNCOG and simulation. The first method shown is EPL, the second is BNCOG and the last is the result of the framework proposed in this work. It can be observed the simulation results are the same as the offered by state-of-the-art algorithms, which for the proposed network are the exact worst-case results, verifying the correctness of the simulation models.

B. NETWORK USE CASE B

It has been assessed in the previous use case that the proposed framework implements properly the network behaviors, with its results matching those offered by the baseline mechanisms. An additional use case is defined in order to evaluate execution times and the level of detail provided by the framework. This includes many different path lengths and several routes where the number of VLs is increased in comparison with the previous use case. With the increased number of VLs the number of messages sent during the simulation also grows, which is the major impact factor for the simulation execution time.

This second network is simulated with the configuration summarized in TABLE 5. It has been established a small difference between the minimum and maximum packet size in order to obtain variation in the traces. These have been set around 200 bytes being it a common value in AFDX networks for the configured BAG of 4ms.

From this it can be observed how the level of detail provided by the AVorks goes far beyond of what can be achieved using Network Calculus approaches. Packet traces obtained from the simulation can be used to acquire metrics as time series. This is an important feedback for the design process as it contains information during normal operation, which can be used to evaluate the performance of the network

TABLE 5. Network setup for the use case B.

Transmitter	VLs	Receiver	Route	Packet Size	BAG
ES1	VL1-4	ES5	$ES1 \rightarrow S3 \rightarrow S4 \rightarrow ES5$	150-200 B	4 ms
ES1	VL5-8	ES6	$ES1 \rightarrow S3 \rightarrow S6 \rightarrow ES6$	150-200 B	4 ms
ES1	VL9-12	ES3	$ES1 \rightarrow S3 \rightarrow S5 \rightarrow ES3$	150-200 B	4 ms
ES2	VL35-36	ES7	$ES2 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow ES7$	50-150 B	4 ms
ES2	VL37-38	ES6	$ES2 \rightarrow S2 \rightarrow S3 \rightarrow S5 \rightarrow S6 \rightarrow ES6$	50-150 B	4 ms
ES2	VL39-40	ES3	$ES2 \rightarrow S12 \rightarrow S3 \rightarrow S5 \rightarrow ES3$	50-150 B	4 ms
ES3	VL13-20	ES1	$ES3 \rightarrow S2 \rightarrow S3 \rightarrow ES1$	100-200 B	4 ms
ES4	VL29-30	ES7	$ES4 \rightarrow S1 \rightarrow S4 \rightarrow ES7$	50-100 B	4 ms
ES5	VL31-32	ES7	$ES5 \rightarrow S1 \rightarrow ES7$	400-600 B	4 ms
ES6	VL33-34	ES7	$ES6 \rightarrow S5 \rightarrow S1 \rightarrow S4 \rightarrow ES7$	50-150 B	4 ms
ES7	VL21-22	ES5	$ES7 \rightarrow S4 \rightarrow ES5$	150-200 B	4 ms
ES7	VL23-24	ES4	$ES7 \rightarrow S4 \rightarrow S1 \rightarrow ES4$	150-200 B	4 ms
ES7	VL25-26	ES6	$ES7 \rightarrow S4 \rightarrow S1 \rightarrow S6 \rightarrow ES6$	150-200 B	4 ms
ES7	VL27-28	ES2	$ES7 \rightarrow S4 \rightarrow S3 \rightarrow S2 \rightarrow ES2$	150-200 B	1 ms

and establish figure of merits for the design beyond the worst-case delays used for certification.

Accordingly, delay metrics are used to obtain different statistical measures. The statistics from the delays of the simulated scenario for each VL are shown in TABLE 6. From these statistics, it can be assessed that VLs of each “group” (i.e., those between the same two ESs) have similar behaviors, since within groups the configuration and paths are the same. Conversely, comparing different groups such as VLs 21-22 and VLs 33-34, the different number of nodes in the path impacts the delay doubling it for VLs 33-34 even if having half average packet size (as shown in TABLE 5). The difference in the number of nodes in the path does not only increase the delay, but also its standard deviation, as the queuing delay of a VL is affected by the transmission time of the packets. Furthermore, it can be seen that the difference of the minimum a maximum packet size increases the standard deviation of the delay comparing groups VLs 31-32 and VLs 21-22. Here, the difference between maximum and minimum in VLs 31-32 is 4 times the difference in VL 21-22, and the former has roughly 4 times the standard deviation of the latter.

Similarly, traces can be also used to obtain throughput statistical measures. An example of these measures is shown in TABLE 7. In the case of throughput, given the fact that a packet is transmitted periodically according to the established BAG (also for 4ms), the bigger the packet is, the higher the throughput. In the same way, a shorter periodicity would increase the throughput as well. Furthermore, it can be seen that the variance of the throughput increases as the difference between minimum and maximum size increases.

For the analysis of the computational costs of the framework, the framework is tested for the use case B using a PC platform equipped with an AMD Ryzen 5 3600 CPU and 4x8GB of DDR4 RAM. During the evaluation of a model the BAG is considered the most impacting parameter as it is directly related to the total number of messages sent, making

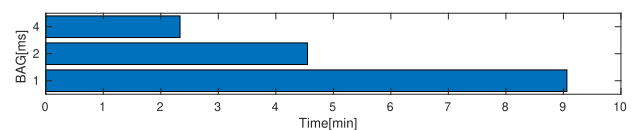


FIGURE 8. Execution times.

the same functions of the model to be executed more times. This would be similar to changing to a topology with more switches in the path of each VL.

In this way, as represented in FIGURE 8, the execution time measurements are performed for three different simulations of one second of simulated time and setting the BAG of every VL to 1 ms, 2 ms and 4 ms. This means a total of 80000, 40000 and 20000 messages sent and reaching around 2, 4 and 8 minutes of execution time respectively. It can be observed that this time is linearly dependent with the number of sent messages. In comparison to the results in [42], where sending 3500 messages took 10 minutes using an Intel Core(TM) i5-2540M CPU at 2.60 GHz with 8 GB RAM the results of the framework in this work are promising, although there are some differences in the network topology simulated and hardware differences are noticeable (hardware used in the present work is about twice as fast in single core speed). However, these execution times would allow evaluating real networks in a timely manner. Furthermore, from the observed linearity of the time with the number of messages it can be expected that the execution time will not grow much more rapidly than the number of VLs.

C. NETWORK USE CASE C

The previous use cases were used to ensure the simulator behaves correctly and the simulation times are adequate to simulate real aircraft-like avionics networks. Then, it is necessary to assess the utility of the framework in the development and testing of new network elements. For this purpose, a key network example modelling an Audio Communications Manager (ACM), two Digital Audio Control

TABLE 6. Summary of statistical measures for delays (in microseconds).

VL	Mean	Median	Std	VL	Mean	Median	Std	VL	Mean	Median	Std
1	78.66	78.68	3.46	15	74.48	74.48	6.01	29	55	55.28	3.5
2	79.49	79.76	3.02	16	74.81	75.44	6.01	30	55.72	56.04	3.02
3	80.05	80.48	3.15	17	74.78	75.56	5.82	31	100.86	101.28	8.93
4	80	80	2.91	18	75.52	75.68	5.44	32	100.54	101.04	8.11
5	78.79	79.16	3.64	19	75.1	75.8	5.7	33	85.6	85.12	9.71
6	80.01	80.28	3.22	20	75.02	75.44	5.99	34	88.94	89.6	8.02
7	80.11	80.24	3.01	21	47.26	47.2	2.3	35	87.06	87.2	9.47
8	80	80.24	2.78	22	47.58	47.52	2.19	36	90.38	91.2	7.51
9	78.48	78.2	3.5	23	78.6	78.56	3.62	37	112	110.8	10.82
10	79.66	80.04	3.31	24	79.71	80	3.03	38	115.96	117.6	10.39
11	79.51	79.68	2.97	25	111.65	112	4.45	39	88.24	88.52	7.88
12	79.85	79.88	2.91	26	112.53	113.2	3.78	40	88.89	89.24	7.64
13	72.67	73.28	6.69	27	110.25	110.08	4.8				
14	74.07	74.32	6.23	28	111.79	112.48	4.28				

TABLE 7. Summary of statistical measures for throughput (in kbps).

VL	Mean	Median	Std	VL	Mean	Median	Std	VL	Mean	Median	Std
1	388.49	388.63	28.84	15	339.17	340.29	56.94	29	191.59	193.97	29.14
2	388.13	388.31	28.94	16	339.34	343.26	57.66	30	189.15	190.02	29.56
3	393.49	396.36	30.81	17	337.51	335.83	56.87	31	1057.21	1065.32	115.29
4	389.51	390.4	29.69	18	343.43	348.99	58.33	32	1039.72	1042.88	110.42
5	390.79	393.87	30.43	19	341.88	346.27	56.98	33	235.49	232.46	60.82
6	392.48	389.98	30.12	20	337.99	334.39	61.15	34	239.38	234.56	56.15
7	391.45	391.35	29.8	21	390.98	390.2	28.78	35	244.07	244.96	59.19
8	390.13	390.25	28.26	22	390.55	389.4	30.08	36	243.03	242.96	58.07
9	387.67	385.35	29.15	23	387.41	387.09	30.08	37	235.95	230.09	57.34
10	390.58	388.02	30.34	24	389.64	390.26	29	38	238.96	239.97	58.57
11	384.95	381.98	28.99	25	396.52	400.6	29.6	39	238.8	240.78	56.96
12	390.15	391.87	29.34	26	389.97	389.97	30.01	40	232.36	230.76	58.41
13	338.11	343.2	55.61	27	389.58	388.5	30.06				
14	334.75	334.88	59.27	28	387.49	385.66	30.13				

TABLE 8. Network setup for the use case C.

Transmitter	VLS	Receiver	Route	Packet Size	BAG	Transmission start time
ACM	VL1	DACP1	ACM → S3 → S4 → DACP1	12 B	1 ms	1/7 ms
ACM	VL2	DACP2	ACM → S3 → S6 → DACP2	12 B	1 ms	1/7 ms + t_{ixVL1}
ACM	VL3	SDR	ACM → S3 → S5 → SDR	40 B	50 ms	1/7 ms + t_{ixVL2}
ES2	VL12	CLOCK	ES2 → S2 → S3 → S4 → CLOCK	54 B	1 ms	0 ms
ES2	VL13	DACP2	ES2 → S2 → S3 → S5 → S6 → DACP2	400 B	1 ms	0 ms + t_{ixVL12}
ES2	VL14	SDR	ES2 → S12 → S3 → S5 → SDR	400 B	1 ms	0 ms + $\sum_{i=12}^{13} t_{ixVLi}$
SDR	VL4	ACM	SDR → S2 → S3 → ACM	60 B	50 ms	2/7 ms
ES4	VL9	CLOCK	ES4 → S1 → S4 → CLOCK	54 B	1 ms	3/7 ms
DACP1	VL10	CLOCK	DACP1 → S1 → CLOCK	54 B	1 ms	4/7 ms
DACP2	VL11	CLOCK	DACP2 → S5 → S1 → S4 → CLOCK	54 B	1 ms	5/7 ms
CLOCK	VL5	DACP1	CLOCK → S4 → DACP1	44 B	1 ms	6/7 ms
CLOCK	VL6	ES4	CLOCK → S4 → S1ES4	44 B	1 ms	6/7 ms + t_{ixVL5}
CLOCK	VL7	DACP2	CLOCK → S4 → S1 → S6 → DACP2	44 B	1 ms	6/7 ms + $\sum_{i=5}^6 t_{ixVLi}$
CLOCK	VL8	ES2	CLOCK → S4 → S3 → S2 → ES2	44 B	1 ms	6/7 ms + $\sum_{i=5}^7 t_{ixVLi}$

Panel (DACP), two Software Defined Radio (SDR) and two ES with the addition of time synchronization using an ES as master clock to enhance determinism is proposed.

The ACM is the equipment responsible of sending the corresponding setting to the SDRs and the Digital Audio Control Panels (DACPs), as well as receiving the status

responses from the SDRs. However, as the DACPs are reception-only equipment, these can be modelled using an ES. The ACM allows the crew to control de audio communication through a Human-Machine Interface (HMI). This makes the traffic sent by the ACM asynchronous, but transmitted in a slot determined by the BAG of the VL.

Regarding the simulation model, the ACM is based on the ES model, but introducing a new packet generator, which generates packets with a BAG of 50 ms and when a randomly generated signal coming from an HMI is received. Also, the SDR is based on the ES model, but replacing the packet generator with one that generates packet with a BAG of 50 ms.

The network topology and parameters used for this simulation are presented in TABLE 8 and two experiments will be carried out with this setup. In this configuration, time synchronization between elements is supposed using a protocol such as Precision Time Protocol (PTP). This is represented by the 40 B packets sent by the clock and the 54 B answers packets sent by the other elements. Furthermore, the 12 B packet sent by the ACM are the DACP settings, while the 40 B packets are the SDR settings. In the same fashion, the SDR reports its status to the ACM with 60 B packets. Both sends their packets every 50 ms, and the ACM sends the packets to the DACPs randomly, but using 1 ms intervals. Lastly, ES2 transmit to the DACP2 and SDR packets every 1 ms with a packet size of 400 B. However, in a second experiment the packet size of ES is changed to 600 B.

The results of the two experiments are shown in FIGURE 9. The experiments are carried out using the global time synchronization to enhance the determinism. In this case, all the ESs' transmission start times are equidistantly distributing within 1 ms. In FIGURE 9a can be seen that for this configuration delays are constant. However, in FIGURE 9b can be seen that if the size of the packet transmitted by ES2 is increased to 600 B, the delay of VL14 shows periodic peaks. These are caused by the packets transmitted by the ES now temporally overlapping with the packets transmitted by the SDR in a switch. In this way, while the VL13 is being transmitted, the SDR packet arrives at the queue and the glsVL14 has to wait for the SDR packet to be transmitted, resulting in an increase of delay every 50 ms when the packet from the SDR is transmitted.

VI. DISCUSSION AND RESEARCH CHALLENGES FOR AVIONICS NETWORKS

In the previous sections, the proposed AVorks framework has shown its capabilities to provide support the V&V tasks for AFDX networks. However, looking at the general field of avionics communications and as network requirements grow to be more and more demanding in multiple scenarios, current protocols need to be more efficient and reliable.

As AFDX does not consider congestion defects caused by Ethernet, the event-driven network paradigm is changing to new standards based on optimized communications for safety-critical systems, which are expected to be the ones generally implemented in avionics. Among these specifications are TSN [56] and TTEthernet [5].

TSN is born as an evolution of the Ethernet-AVB standard, in which an additional traffic flow known as *Control Data Traffic* is included, being optimized for real-time oriented

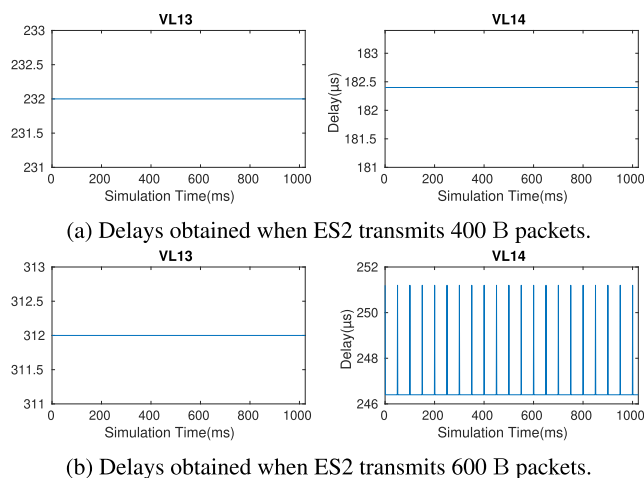


FIGURE 9. Delays obtained from the simulation for VLS 13 and 14.

applications in a strict way. To implement these changes, it is necessary to introduce a Time Aware Shaping (TAS) under conditions of maximum reliability and reduced latency [57]. In addition, to ensure reliability and low latency, this standard requires a synchronization process, so that the endpoint can calculate time parameters such as delays and jitter based on time stamps [5]. Due to these features, although TSN has not yet made its appearance in the aeronautical communications systems, although it is expected to start being used in future aircraft generation [56].

As an intermediate step, and supported by the proposed AVorks framework, some of the mechanisms that could be evaluated for AFDX are the global synchronization and the TAS/Burst Limiting Shaper (BLS) used in TSN [57]. With TAS all devices are synchronized and reserved bandwidth cannot be used for other traffic. Moreover, BLS ensures resources by limiting bandwidth and using priorities. The use of this synchronization and shapers aims at reducing worst-case end-to-end delays and increase the network capacity as well as highly increase the capacity, while maintaining the reliability of the avionics networks.

Furthermore, hybrid networks using several standards can easily be part of a deployment, combining both legacy and newly defined standards in order to cover the requirements of different final elements and/or traffics. As these solutions could be more cost-effective, their assessment is a challenge.

In order to address the presented challenges in terms of new standards, network services and testing and optimization techniques, proper platforms and tools are required. Here, the proposed AVorks is considered a key step to build upon in order to evaluate these new protocols and the interoperability of hybrid networks.

The proposed framework is applicable to develop and evaluate the expected standards and mechanisms. The adopted MBSE approach is especially suitable to facilitate the development of novel algorithms, as existing blocks can serve as a base for the new models. Moreover, the proposed framework, as shown in Section IV allows for the type

of iterative testing required in aeronautical communications systems. Also, the described AVorks specific implementation, being based on Simulink, allows the models to be tested via simulations and then straightforwardly deployed in development boards for further evaluation with real elements as HIL [14]. In this way, the simulations can be used to verify and validate the specifications of the proposed network, but also allow to develop and test custom software that can be used in hardware after the network design phase. This is especially relevant in aeronautical communications, where both a progressive evolution of the systems as well as an exhaustive testing are necessary due to their complexity and reliability requirements.

Current V&V frameworks in avionics are limited to evaluate proposed topologies using a pre-established set of tests. However, defining such tests and re-designing the network topology is slow and costly. In this area, EASA has defined a technology roadmap for the introduction of Artificial Intelligence (AI) in aviation [58]. In this way, AI solutions coming from other research fields are expected to be implemented in V&V frameworks, such as testing software completeness [59] and network topology optimization using genetic algorithms [60], in order to significantly accelerate this process and reduce cost. Here, the proposed framework can support the application of AI in avionics by offering an environment to implement the algorithms to automatically evaluate and optimize the network during the design phase.

VII. CONCLUSION AND OUTLOOK

This paper has proposed a framework for the verification and validation of avionics networks based both in simulation models and network calculus. This framework has been focused on AFDX protocol-based implementations, given therefore the present work a summary of the key principles and characteristics of said protocol. The developed framework includes the model generation mechanisms and algorithms to evaluate any given AFDX network, in order to assess different network topologies and configurations.

In the presented evaluation, the developed tool is focused on the calculation of data link layer network metrics, which will support the verification and validation of the network requirements as well as optimizations of the configuration under analysis. Hence, the defined AVorks framework has been tested against state-of-the-art Network Calculus applications obtaining consistent results in estimating worst-case end-to-end delays, while providing a much deeper level of detail. Hence, by comparing the results with an exact method to estimate worst-case end-to-end delays, the correctness of the modelling of the different blocks has been verified. Moreover, analyzing the execution times of a realistic aircraft-like network scenario simulation, the appropriateness of this framework to be used in widely available computer platforms has been demonstrated. In this way, the provided results have shown the capabilities of the proposed framework in the V&V of avionics networks.

REFERENCES

- [1] H. Butz, "Open integrated modular avionic (IMA): State of the art and future development road map at airbus deutschland," *Signal*, vol. 10, p. 1000, Mar. 2008.
- [2] FAA and AIR-600, *25-7D—Flight Test Guide for Certification of Transport Category Airplanes*, FAA (Federal Aviation Administration), Washington, DC, USA, 2018.
- [3] G. Gratton, *Initial Airworthiness: Determining the Acceptability of New Airborne Systems*. Cham, Switzerland: Springer, 2018.
- [4] *IEEE Standard for Local and metropolitan area networks—Audio Video Bridging (AVB) Systems*, IEEE Standard 802.1BA-2011, 2011, pp. 1–256.
- [5] A. T. Loveless, "On TTEthernet for integrated fault-tolerant spacecraft networks," in *Proc. AIAA SPACE Conf. Expo.*, Aug. 2015, p. 4526.
- [6] S. Kazi, "Architecting of avionics full duplex Ethernet (AFDX) aerospace communication network," *Int. J. Electron. Commun. Technol. (IJEET)*, vol. 4, pp. 1–5, Jun. 2013.
- [7] Aeronautical Radio, *ARINC 664 P7, Aircraft Data Network Part 7 Avionics Full-Duplex Switched Ethernet Network*, ARINC Industry Activities (ARINC), Annapolis, MD, USA, 2009.
- [8] *IEEE Standard for Ethernet*, IEEE Standard 802.3-2018, 2018.
- [9] F. A. Administration, *Software Considerations in Airborne Systems and Equipment Certification*, document RTCA/DO-178C, 2012.
- [10] *Design Assurance Guidance for Airborne Electronic Hardware*, document RTCA/DO-254, 2005.
- [11] T. Gaska, C. Watkin, and Y. Chen, "Integrated modular avionics—past, present, and future," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 9, pp. 12–23, Sep. 2015.
- [12] D. N. Darwesh, B. Annighofer, and R. Reichel, "A demonstrator for the verification of the selective integration of the flexible platform approach into integrated modular avionics," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2018, pp. 1–10.
- [13] L. Li, N. L. Soskin, A. Jbara, M. Karpel, and D. Dori, "Model-based systems engineering for aircraft design with dynamic landing constraints using object-process methodology," *IEEE Access*, vol. 7, pp. 61494–61511, 2019.
- [14] M. Lizarraga, V. Dobrokhodov, G. Elkaim, R. Curry, and I. Kaminer, "Simulink based hardware-in-the-loop simulator for rapid prototyping of UAV control algorithms," in *Proc. AIAA Infotech@Aerosp. Conf.*, 2012, p. 1843. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2009-1843>
- [15] J. Yao, G. Zhu, X. Liu, and A. Jou, "Optimal bandwidth allocation for non-critical traffics in AFDX network," in *Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jul. 2012, pp. 1727–1733.
- [16] A. Al Sheikh, O. Brun, M. Chéramy, and P.-E. Hladik, "Optimal design of virtual links in AFDX networks," *Real-Time Syst.*, vol. 49, no. 3, pp. 308–336, May 2013.
- [17] D. An, K. H. Kim, and K.-I. Kim, "Optimal configuration of virtual links for avionics network systems," *Int. J. Aerosp. Eng.*, vol. 2015, Nov. 2015, Art. no. 329820.
- [18] A. Jouy, J. Yao, and G. Zhu, "Optimal bandwidth allocation with dynamic multi-path routing for non-critical traffic in AFDX networks," in *Proc. 20th IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2014, pp. 600–607.
- [19] H. Ayed, A. Mifdaoui, and C. Fraboul, "Hierarchical traffic shaping and frame packing to reduce bandwidth utilization in the AFDX," in *Proc. 9th IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Jun. 2014, pp. 77–86, pp. 77–86.
- [20] B. Annighofer, "Network topology optimization for distributed integrated modular avionics," in *Proc. IEEE/AIAA 33rd Digit. Avionics Syst. Conf. (DASC)*, Oct. 2014, pp. 4A1–1.
- [21] M. Li, M. Lauer, G. Zhu, and Y. Savaria, "Determinism enhancement of AFDX networks via frame insertion and sub-virtual link aggregation," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1684–1695, Aug. 2014.
- [22] X. Li, N. Huang, and F. Zhao, "A genetic algorithm based configuration optimization method for AFDX," in *Proc. 10th Int. Conf. Rel., Maintainability Saf. (ICRMS)*, Aug. 2014, pp. 440–444.
- [23] J. Yao, J. Wu, Q. Liu, Z. Xiong, and G. Zhu, "System-level scheduling of mixed-criticality traffics in avionics networks," *IEEE Access*, vol. 4, pp. 5880–5888, 2016.
- [24] S. Gurjar and B. Lakshmi, "Optimal scheduling policy for jitter control in AFDX end-system," in *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, May 2014, pp. 1–4.

- [25] A. Finzi and S. Craciunas, "Breaking vs. solving: Analysis and routing of real-time networks with cyclic dependencies using network calculus," in *Proc. RTNS*, Nov. 2019, pp. 101–111.
- [26] J.-Y. L. Boudec and P. Thiran, *Network Calculus, a Theory of Deterministic Queuing Systems for the Internet*. Berlin, Germany: Springer, 2012.
- [27] M. Boyer and C. Fraboul, "Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus," in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, May 2008, pp. 11–20.
- [28] G. Kemayo, N. Benammar, F. Ridouard, H. Bauer, and P. Richard, "Improving AFDX end-to-end delays analysis," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–8.
- [29] Q. Xu and X. Yang, "Performance analysis on transmission estimation for avionics real-time system using optimized network calculus," *Int. J. Aeronaut. Space Sci.*, vol. 20, no. 2, pp. 506–517, Jun. 2019.
- [30] X. Li, O. Cros, and L. George, "The trajectory approach for AFDX FIFO networks revisited and corrected," in *Proc. IEEE 20th Int. Conf. Embedded Real-Time Comput. Syst. Appl.*, Aug. 2014, pp. 1–10.
- [31] S. Liu, F. He, T. Wang, and Y. Li, "Optimization of trajectory approach in end-to-end delay analysis considering the flow offsets scheduling," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2016, pp. 3614–3620.
- [32] J.-L. Scharbag and C. Fraboul, "A generic simulation model for end-to-end delays evaluation on an avionics switched Ethernet," *IFAC Proc. Volumes*, vol. 40, no. 22, pp. 159–166, 2007.
- [33] H. Charara and C. Fraboul, "Modelling and simulation of an avionics full duplex switched Ethernet," in *Proc. Adv. Ind. Conf. Telecommun./Service Assurance Partial Intermittent Resour. Conf./E-Learn. Telecommun. Workshop (AICT/SAPIR/ELETE)*, 2005, pp. 207–212.
- [34] D. Song, X. Zeng, L. Ding, and Q. Hu, "The design and implementation of the AFDX network simulation system," in *Proc. Int. Conf. Multimedia Technol.*, Oct. 2010, pp. 1–4.
- [35] B. Annighoefer, H. Ihle, and F. Thielecke, "An easy-to-use real-time AFDX simulation framework," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2016, pp. 1–9.
- [36] J. Fernández, H. Pérez, J. Javier Gutiérrez, and M. G. Harbour, "AFDX emulator for an ARINC-based training platform," in *Reliable Software Technologies—Ada-Europe*, J. A. de la Puente and T. Vardanega, Eds. Cham, Switzerland: Springer, 2015, pp. 212–227.
- [37] H. Charara, J.-L. Scharbag, J. Ermont, and C. Fraboul, "Methods for bounding end-to-end delays on an AFDX network," in *Proc. 18th Euromicro Conf. Real-Time Syst. (ECRTS)*, 2006, p. 10.
- [38] Q. Yang, H. Lu, and X. Tu, "Simulation and experiment of AFDX network based on OMNeT++," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 5849–5854.
- [39] X. Liu, Z. Du, and K. Lu, "Modeling and simulation of avionics full duplex switched Ethernet (AFDX network) based on OPNET," in *Proc. 3rd Joint Int. Inf. Technol., Mech. Electron. Eng. Conf. (JIMEC)*, 2018, pp. 307–310.
- [40] N. E.-D. Safwat, A. Zekry, and M. Abouelatta, "Avionics full-duplex switched Ethernet (AFDX): Modeling and simulation," in *Proc. 32nd Nat. Radio Sci. Conf. (NRSC)*, Mar. 2015, pp. 286–296.
- [41] J. Xia, W. Yuan, and R. Bai, "Study on real-time performance of AFDX using OPNET," in *Proc. Int. Conf. Control, Autom. Syst. Eng. (CASE)*, Jul. 2011, pp. 1–5.
- [42] M. Ashjaei, M. Behnam, and T. Nolte, "SEtSim: A modular simulation tool for switched Ethernet networks," *J. Syst. Archit.*, vol. 65, pp. 1–14, Apr. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1383762116000230>
- [43] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén, "How does control timing affect performance? Analysis and simulation of timing using Jitterbug and truetype," *IEEE Control Syst. Mag.*, vol. 23, no. 3, pp. 16–30, Jun. 2003.
- [44] Aeronautical Radio, *ARINC 653P1-3 Avionics Application Software Standard Interface, Part 1, Required Services*, ARINC Industry Activities (ARINC), Annapolis, MD, USA, Nov. 2010.
- [45] H. Strauch, K. Luig, and S. Bennani, "Model based design environment for launcher upper stage GNC development," in *Proc. Workshop Simul. Eur. Space Programmes (SESP)*, Mar. 2015. [Online]. Available: <https://indico.esa.int/event/93/contributions/3565/>
- [46] H. Daigmorte, P. D. Saqui-Sannes, and R. Vingerhoeds, "A SysML method with network dimensioning," in *Proc. Int. Symp. Syst. Eng. (ISSE)*, Oct. 2019, pp. 1–8.
- [47] L. Fernandez-Olmos, F. Burrull, and P. Pavon-Marino, "Net2Plan-AFDX: An open-source tool for optimization and performance evaluation of AFDX networks," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2016.
- [48] M. Kos and Z. Vrdoljak, "Simulation of ATM network behavior with the event driven ATM simulator," in *Proc. 10th Medit. Electrotech. Conf. Inf. Technol. Electrotechnol. Medit. Countries (MeleCon)*, May 2000, pp. 202–205.
- [49] F. Brajou and P. Ricco, "The airbus A380—An AFDX-based flight test computer concept," in *Proc. AUTOTESTCON*, 2004, pp. 460–463.
- [50] Aeronautical Radio, *ARINC 429 P1-17*, ARINC Industry Activities (ARINC), Annapolis, MD, USA, 2004.
- [51] N. El-DinSafwat, M. A. El-Dakrouy, and A. Zekry, "The evolution of aircraft data networks," *Int. J. Comput. Appl.*, vol. 94, no. 11, pp. 27–32, May 2014.
- [52] T. Liew, *Network Traffic Control and Bandwidth Allocation*. Hoboken, NJ, USA: Wiley, 2010, ch. 8, pp. 139–171.
- [53] D. Medhi and K. Ramasamy, "Traffic conditioning," in *Network Routing* (The Morgan Kaufmann Series in Networking), D. Medhi and K. Ramasamy, Eds., 2nd ed. Boston, MA, USA: Morgan Kaufmann, 2018, Ch. 18, pp. 626–644. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780128007372000211>
- [54] J. Ermont and C. Fraboul, "Modeling a spacewire architecture using timed automata to compute worst-case end-to-end delays," in *Proc. IEEE 18th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2013, pp. 1–4.
- [55] P. Munier, *Polyspace*. Hoboken, NJ, USA: Wiley, 2013, ch. 4, pp. 123–153. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118561829.ch4>
- [56] IEEE. (2018). *Time-Sensitive Networking (TSN) Task Group*. [Online]. Available: <https://1.ieee802.org/tsn/>
- [57] A. Finzi, "Specification and analysis of an extended AFDX with TSN/BLS shapers for mixed-criticality avionics applications," Ph.D. dissertation, Dept. Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Univ. Toulouse, Toulouse, France, Jun. 2018.
- [58] EASA. (2020). *Artificial Intelligence Roadmap: A Human-Centric Approach to AI in Aviation*. [Online]. Available: <https://www.easa.europa.eu/sites/default/files/dfu/EASA-AI-Roadmap-v1.0.pdf>
- [59] N. Iqbal and J. Sang, "Fuzzy logic testing approach for measuring software completeness," *Symmetry*, vol. 13, no. 4, p. 604, Apr. 2021. [Online]. Available: <https://www.mdpi.com/2073-8994/13/4/604>
- [60] P.-L. Bourbonnais, C. Morency, M. Trépanier, and E. Martel-Poliquin, "Transit network design using a genetic algorithm with integrated road network and disaggregated O-D demand data," *Transportation*, vol. 48, pp. 95–130, Aug. 2021.



JAVIER VILLEGAS received the degree in telecommunications systems engineering from the University of Málaga, Spain, where he is currently pursuing the dual M.Sc. degree in telecommunication engineering and in telematic engineering. He is working as a Research Assistant with the Department of Communications Engineering, University of Málaga.



SERGIO FORTES (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Spain. He began his career in the field of satellite communications and holding positions in European space agencies, where he has participated in various research and consultant activities on broadband and aeronautical satellite communications. In 2012, he joined the University of Málaga, where his research is focused on self-organizing networks for cellular communications.



VICENTE ESCAÑO has been working at AERTEC, since 2016, focusing on avionics embedded systems for safety critical systems and systems communications for integrated modular avionics (IMA) platforms. He has participated in many of the key commercial projects developed at AERTEC Aerospace and Defence area for clients, such as airbus defense and space and airbus operations in three countries: Spain (Sevilla and Getafe), France (Toulouse), and Germany (Finkenwerder and Buxtehude).



BENJAMÍN COLOMER received the M.Sc. degree in telecommunications engineering from the University of Navarra, in 2010, and the master's degree in business administration, in 2012. He is currently a Senior Engineer. He has been working at AERTEC, since February 2011, started as part of the Industrial Systems Engineering Team. He leads the Model-Based Design Team, AERTEC. He also acted as the Project Manager of Aerospace and Defense Area. He is involved in several project for aeronautical environment as development of industrial means, electronic solutions, and activities using tools as the MATLAB/Simulink.



CARLOS BAENA received the M.Sc. degree in telematic engineering from the University of Málaga, Spain. He is currently pursuing the Ph.D. degree with the Department of Communications Engineering. Since 2018, he has been working as a Research Assistant with the Department of Communications Engineering.



RAQUEL BARCO received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Spain. From 1997 to 2000, she worked at Telefónica, Madrid, and the European Space Agency, Darmstadt, Germany. In 2000, she joined the University of Málaga, where she is currently an Associate Professor. She has worked in projects with major mobile communications operators and vendors. She is an author of more than 100 high-impact journal and conference articles.

...