Abstract—In this paper, we propose an extension of the AFDX standard, incorporating a TSN/BLS shaper, to homogenize the avionics communication architecture, and enable the interconnection of different avionics domains with mixed-criticality levels, e.g., legacy AFDX traffic, Flight Control and In-Flight Entertainment. First, we present the main specifications of such a proposed solution. Then, we detail the corresponding worst-case timing analysis, using the Network Calculus framework, to infer real-time guarantees. Finally, we conduct the performance analysis of such a proposal on a realistic AFDX configuration. Results show the efficiency of the Extended AFDX standard to noticeably enhance the medium priority level delay bounds, while respecting the higher priority level constraints, in comparison with the legacy AFDX standard.

I. INTRODUCTION

The growing number of interconnected end-systems and the expansion of exchanged data in avionics have led to an increase in complexity of the communication architecture. To cope with this trend, a first communication solution based on a high rate backbone network, i.e., the AFDX (Avionics Full Duplex Switched Ethernet) [1], has been implemented by Airbus in the A380, to interconnect critical subsystems. More recently, some low rate data buses, e.g., CAN [12], have been introduced in the A350 and the A400M to handle some specific avionics domains, such as the I/O process and the Flight Control Management. Although this architecture reduces the increase in complexity of the communication architecture, it conjointly leads to inherent heterogeneity and new challenges to guarantee the real-time requirements.

To cope with these emerging issues, with the maturity and reliability progress of the AFDX after a decade of successful use, a homogeneous avionic communication architecture based on such a technology to interconnect different avionics domains may bring significant advancements, such as quick installation and maintenance and reduced weight and costs. Furthermore, this new communication architecture needs to support, in addition to the legacy AFDX traffic profile, called Rate Constrained (RC) traffic, at least two extra profiles. The first, denoted by Safety-Critical Traffic (SCT), is specified to support flows with hard real-time constraints and the highest priority level, e.g., flight control data; whereas the second is for Best-Effort (BE) flows with no delivery constraint and the lowest priority level, e.g., In-Flight Entertainment traffic. It is worth noting that such a priority assignment depending on the traffic criticality is usually used in avionics to facilitate the certification process.

Hence, an ultimate avionic communication architecture has to fulfill the following necessary conditions:

| Condition | Description |
|-----------|-------------|
| 1         | keeping the low (re)configuration effort and high modularity level, guaranteed by the current AFDX standard; |
| 2         | supporting mixed-criticality applications while guaranteeing the predictability requirement, i.e., the existence of a bounded latency for each traffic flow; |
| 3         | enforcing the Quality of Service (QoS) features while limiting the impact of the highest priority traffic on the legacy AFDX traffic. |

There are many existing solutions to support mixed-criticality applications in embedded systems, and particularly in avionics [17] [22] [19] [23] [3]. However, each one satisfies some of the aforementioned conditions better than others, but there is no solution meeting all of them.

Therefore, our main contributions in this paper are threefold: (i) first, the specification of an extended AFDX standard satisfying the three aforementioned conditions, based on the Burst-Limiting Shaper (BLS) [5] defined in Time Sensitive Networking (TSN) task group [22], in Section III; (ii) second, an appropriate system modeling and timing analysis, based on the Network Calculus framework [8], to evaluate its impact on SCT and RC guarantees in Sections IV and V; (iii) third, in Section VI, the validation of such a proposal in the case of a realistic avionics network, interconnecting more than 60 end-systems and varying the maximum utilisation rate of SCT and RC traffic, to prove its efficiency to guarantee SCT traffic constraints, while enhancing the RC guarantees.

II. RELATED WORK

In this section, we review the most relevant solutions to support mixed-criticality applications in embedded systems, and the main worst-case timing analyses of the TSN/BLS shaper.

A. Supporting Mixed-criticality Applications in Embedded Systems

Various solutions have been proposed in the literature to support mixed-criticality applications in embedded systems

...
and particularly in avionics [18]. These solutions can be actually categorised according to the implemented communication paradigm, i.e., mainly event-triggered or time-triggered. This parameter is of utmost importance to quantify the reconfiguration effort needed by the alternative avionics communication architecture, in comparison to the current AFDX standard (which is an event-triggered solution). Furthermore, it conditions the modularity level of the selected solution. The event-triggered paradigm is known as highly flexible and facilitates the system reconfiguration, but it infers at the same time an indeterminism level and needs further proofs to verify the predictability requirement. On the other hand, the time-triggered paradigm is highly predictable, but presents some limitations in terms of system reconfigurability. The considered solutions in this area vs the three specification conditions of an alternative communication architecture, described in Section I, are illustrated in Table I.

| Solutions               | TTE | TAS | PS | BLS | AVB | NP-SP | WRR |
|-------------------------|-----|-----|----|-----|-----|-------|-----|
| refs.                   | [12]| [21]| [19]| [15]| [16]| [23]  |
| Cond. 1                 | X   | X   |    | X   |     |       |
| Cond. 2                 | X   | X   | X  | X   |     |       |
| Cond. 3                 | X   | X   | X  |     |     |       |
|                         |     |     |    | X   | X   |       |

**TABLE I**

**EXISTING SOLUTIONS VS SPECIFICATION CONDITIONS**

The main relevant solutions implementing time-triggered paradigm on top of Switched Ethernet are Time Triggered Ethernet (TTE) [17] and two solutions under standardisation in IEEE 802 Time Sensitive Networking (TSN) Task group [22], i.e., primarily driven by automotive. Both TSN solutions are based on new shapers, implemented on top of Non-Preemptive Static Priority (NP-SP) scheduler: Time-Aware Shaper (TAS) [21] and Peristaltic Shaper (PS) [19]. All of these solutions aim to drastically reduce the jitter of highest priority traffic, using a time scheduler and/or an adequate synchronisation protocol. These facts imply the non fulfilment of both specification conditions 1 and 3, which make these solutions inadequate in our context.

Among the most interesting solutions based on event-triggered paradigm, we distinguish two classes of solutions. The first class is extending the AFDX standard with well-known scheduling schemes, e.g., NP-SP with n priority levels [3] satisfying only conditions 1 and 2 due to the starvation risk for lower priorities, and Weighted-Round-Robin (WRR) [23] meeting only conditions 2 and 3 due to the necessary weights setting, known to be difficult to tune. The second class in this category is integrating credit-based shapers to control generally the highest priority level, in order to limit its impact on lower priority ones and to guarantee real-time communication. This idea has been initiated for Ethernet [7], then applied to Switched Ethernet [3] [11]. More recently, it has been integrated in Ethernet AVB [16]. However, the implemented behaviour in AVB, i.e., the credit becomes negative as soon as one high priority frame is sent, has led to undesirable high latencies for critical traffic. Consequently, this solution satisfies only conditions 1 and 3. On the other hand, there is an interesting solution in TSN group, based on the Burst Limiting Shaper (BLS) [5] on top of NP-SP scheduler [20], which handles the limitations of AVB standard while keeping the event-triggered paradigm of Switched Ethernet. These characteristics meet all the specification conditions. Therefore, it is considered herein as the most interesting solution to be incorporated within the AFDX standard, to enable an homogeneous avionics communication architecture. However, the BLS shaper may lead at the same time to increasing communication latencies for the highest priority traffic, thus requiring real-time constraints verification. Therefore, an appropriate timing analysis to provide worst-case delays has to be considered. An overview of the related work concerning this issue is presented in the next section.

**B. Worst-case Timing Analysis of TSN/BLS Shaper**

There are some interesting approaches in the literature concerning the worst-case timing analysis of TSN network, and more particularly BLS shaper. The first and seminal one in [13] introduces a first service curve model to induce worst-case delay computation. However, this presentation published by the TSN task group has never been extended in a formal paper. The second one has detailed a more formal worst-case timing analysis in [14]. However, this approach has some limitations. Basically, the proposed model does not take into account the impact of either the same priority flows or the higher ones, which will clearly induce optimistic worst-case delays. The last and more recent one in [20] has proposed a formal analysis of TSN/BLS shaper, based on a Compositional Performance Analysis (CPA) method. This approach has handled the main limitations of the model presented in [14]; and interesting results for an automotive case study have been detailed. The impact of BLS on the highest priority traffic has been showed to deteriorate its timing performance, in comparison with a classic NP-SP scheduler.

However, in this paper, our main objective is different from [20] and consists in incorporating BLS in AFDX, denoted as Extended AFDX, to guarantee the highest priority traffic deadline, while limiting its impact on the medium one, i.e., RC. Moreover, our worst-case timing analysis is based on the Network Calculus framework, which has been proved as highly modular and scalable, in comparison with CPA method [15], and very effective to prove the certification requirements of avionics applications [6]. Several existing works have used Network Calculus to analyse the timing performance of Switched Ethernet and AFDX [6] [10] [9] [4]. However, to the best of our knowledge, the issue of modeling and analysing the TSN/BLS on top of a NP-SP scheduler using the Network Calculus has not been handled yet in the literature.

**III. SPECIFICATION OF EXTENDED AFDX**

In this section, we first describe the Extended AFDX switch architecture, implementing the TSN/BLS on top of a NP-SP scheduler. Then, we detail the BLS behaviour and its main parameters.
A. The Extended AFDX Switch

The aim of extending the AFDX switch architecture with the TSN/BLS is to handle mixed criticality data, and more specifically three AFDX traffic profiles, as illustrated in Fig. 1: (i) the Safety-Critical Traffic (SCT) with its priority set by the BLS and the tightest temporal deadline, e.g., Flight-control flows; (ii) Rate Constraint traffic (RC) with the medium priority and a deadline constraint to guarantee, e.g., legacy AFDX flows; (iii) The best-effort traffic (BE) with the lowest priority and no time constraint, e.g., In-Flight Entertainment.

![Extended AFDX switch architecture](image)

- **Input ports**
- **forwarding process**
- **Output ports**

The legacy AFDX standard manages the exchanged data through the Virtual Link (VL) concept. This concept provides a way to reserve a guaranteed bandwidth for each traffic flow. The VL represents a multicast communication, which originates at a single End System and delivers packets to a fixed set of End Systems. Each VL is characterized by: (i) BAG (Bandwidth Allocation Gap), ranging in powers of 2 from 1 to 128 milliseconds, which represents the minimal inter-arrival time between two consecutive frames; (ii) MFS (Maximal Frame Size), ranging from 64 to 1518 bytes, which represents the size of the largest frame sent during each BAG. Furthermore, the legacy AFDX specifies a NP-SP scheduler based on two priority levels within End Systems and switches to enable the QoS features.

In Fig. 1 we illustrate the architecture of the extended AFDX switch. It consists of: (i) store and forward input ports to verify each frame correctness before sending it to the corresponding output port; (ii) a static configuration table to forward the received frames to the correct output port(s) based on their VL identifier; (iii) the output ports with three priority queues, multiplexed with a NP-SP scheduler, and the highest one is shaped with the BLS.

The current AFDX switch distinguishes the flow priority level based on its VL identifier stored in the static configuration table, i.e., for each VL identifier, there is a predefined priority level stored in the table. Hence, to manage both extra AFDX profiles, i.e., SCT and BE, we need to update the configuration table to add the corresponding VL identifiers and their associated priority levels.

Hence, in comparison to the legacy AFDX switch architecture, the main modifications required for the proposed extended AFDX switch consists in: (i) at the software level, updating the static configuration table to manage three priority levels instead of two, Note that the update overhead is very limited since only one additional bit per line is necessary; (ii) at the hardware level, adding an extra priority queue at the output port since the legacy AFDX supports already two priorities, and implementing the BLS for the SCT queue on top of the NP-SP scheduler, as illustrated in Fig. 1.

From the global avionic communication architecture point of view, the extended AFDX standard necessitates the update of the End-System at the application layer to enable a consistent mapping between VL identifiers and the appropriate priority level.

Moreover, the implementation and certification of this extended AFDX may imply extra costs, in comparison with the legacy one. However, this fact is counterbalanced by the major pros of such an homogeneous architecture, in terms of enhancing performance and reducing cables and weight.

B. BLS Shaper

The BLS belongs to the credit-based shapers class. It has been defined in [5] by an upper threshold, \( L_M \), a lower threshold \( L_R \), such as \( 0 \leq L_R < L_M \), and a reserved bandwidth, \( BW \). Additionally, the priority of a class \( j \) shaped by BLS, denoted \( p(j) \), can oscillate between a high and a low value. The low value is usually below the lowest priority of unshaped traffic. In the avionic context, to guarantee the safety isolation level between the different traffic profiles, the low value associated to the SCT is set to be lower than the RC priority level, but higher than the BE's one. Therefore, SCT queue priority oscillates between 0 and 2, RC’s priority is 1 and BE has the priority 3. Thus, when SCT traffic is enqueued, RC is the only traffic that can be sent and it happens when the SCT priority is 2. As a consequence, the BE traffic is isolated from the SCT and RC traffics.

The credit counter varies as follows:

(i) initially, the credit counter starts at 0 and the queue of the burst limited flows is high;
(ii) the main feature of the BLS is the change of priority \( p(j) \) of the shaped queue, which occurs in two contexts: 1) if \( p(j) \) is high and credit reaches \( L_M \); 2) if \( p(j) \) is low and credit reaches \( L_R \);
(iii) when a frame is transmitted, the credit increases (is consumed) with a rate of \( I_{send} \), else the credit decreases (is gained) with a rate of \( I_{idle} \);
(iv) when the credit reaches \( L_M \), it stays at this level until the end of the transmission of the current frame (if any);
(v) when the credit reaches \( L_R \) it stays at this level until the end of the transmission of the current frame (if any). The credit remains at 0 until a new BLS frame is transmitted.

The behaviour of the BLS is illustrated in Fig. 2 where the queue 1 is the shaped queue by the BLS. As shown, the credit is always between 0 and \( L_M \). The different parameters of the BLS shaper are defined as follows:

(i) the decreasing rate is:

\[
I_{idle} = BW \cdot C
\]
where $C$ is the link speed and $BW$ is the percentage of bandwidth reserved for BLS frames.

(ii) the increasing rate is:

$$I_{\text{send}} = C - I_{\text{idle}}.$$ 

Therefore, the implementation of BLS at the hardware level necessitates a counter to keep track of the credit and a timer to handle credit updates. These parameters, i.e., a counter and a timer, induce low extra complexity of implementing a BLS on top of a NP-SP scheduler, in comparison with a regular NP-SP scheduler. It is worth noting that with the BLS, both the priority of the shaped queue and the state of all the queues, i.e., empty or not, define whether the credit is gained or lost. This aspect is depicted in Fig. 2 for two arrival scenarios. The first one (left figure) shows the case of a bursty traffic, where the maximum of traffic shaped by the BLS is sent when its priority is at the low value. Consequently, the second one (right figure) is for sporadic traffic, where we can see that when the queue 1 priority is highest but no frame is available, then the credit is regained. However, when the priority is at the low value and the other queues are empty, then frames can be transmitted and the credit is consumed.

IV. TIMING ANALYSIS METHODOLOGY

We present in this section the worst-case timing analysis methodology based on Network Calculus (NC), and followed to conduct the performance evaluation of our proposed Extended AFDX. We first present the Network Calculus framework and define the necessary and sufficient schedulability conditions. Then, we detail the models of traffic flows, end-systems and switches, which will be extended in the next section to integrate the impact of the BLS on the different traffic classes. Finally, we explain the computation of the upper bounds on end-to-end delays. The main notations used in this paper are presented in Table II where upper indices indicate nodes or components and lower indices indicate traffic classes, flows or priority levels.

A. Network Calculus Framework

The timing analysis detailed in this paper is based on Network Calculus theory [8] providing upper bounds on delays and backlogs. Delay bounds depend on the traffic arrival described by the so called arrival curve $\alpha$, and on the availability of the traversed node described by the so called minimum service curve $\beta$. The definitions of these curves are explained as following.

**Definition 1 (Arrival Curve).** [8] A function $\alpha(t)$ is an arrival curve for a data flow with an input cumulative function $R(t)$, i.e., the number of bits received until time $t$, iff:

$$\forall t, R(t) \leq R \otimes \alpha(t)$$

**Definition 2 (Strict minimum service curve).** [8] The function $\beta$ is the minimum strict service curve for a data flow with an output cumulative function $R^*$, if for any backlogged period $|s, t|$, $R^*(t) - R^*(s) \geq \beta(t - s)$.

**Definition 3 (Maximum service curve).** [8] The function $\gamma(t)$ is the maximum service curve for a data flow with an input cumulative function $R(t)$ and output cumulative function $R^*(t)$ iff:

$$\forall t, R^*(t) \leq R \otimes \gamma(t)$$

The traffic contracts are generally enforced using a leaky-bucket shaper, i.e., the traffic flow is $(r, b)$-constrained and the arrival curve is $\alpha(t) = r \cdot t + b$ for $t > 0$. A common model of service curve is the rate-latency curve $\beta_{R,T}$, defined as $\beta_{R,T}(t) = [R(t) - T]^+$, where $[x]^+$ is the maximum between $x$ and 0.

$$1 \otimes g(t) = \inf_{0 \leq x \leq t} (f(t - x) + g(s))$$

$1 \otimes g(t)$ is called backlogged period if $R(\tau) - R^*(\tau) > 0, \forall \tau \in [s, t]$
Then, to compute the main performance metrics we need the following results.

**Corollary 1.** (Left-over service curve - SP Multiplex)

Consider a system with the strict service $\beta$ and $m$ flows crossing it, $f_1, f_2, ..., f_m$. The maximum packet length of $f_i$ is $l_{i,\max}$ and $f_i$ is $\alpha_i$-constrained. The flows are scheduled by the non-preemptive static priority (NP-SP) policy, where priority of $f_i$ > priority of $f_j \iff i < j$. For each $i \in \{2, ..., m\}$, the strict service curve of $f_i$ is given by:

$$ (\beta - \sum_{j<i} \alpha_j - \max_{k\geq i} l_{k,\max}) \uparrow $$

**Theorem 1** (Performance Bounds).

Consider a flow $F$ constrained by an arrival curve $\alpha$ crossing a system $S$ that offers a minimum service curve $\beta$ and a maximum service curve $\gamma$. The performance bounds obtained at any time $t$ are:

* Backlog $Q$: $\forall t: q(t) \leq v(\alpha, \beta)$
* Delay $D$: $\forall t: d(t) \leq h(\alpha, \beta)$
* Tight Output arrival curve $\alpha^*\!\!(t)$ (Left-over service curve - SP Multiplex)

**Theorem 2.** (Concatenation-Pay Bursts Only Once)

Assume a flow crossing two servers with respective service curves $\beta_1$ and $\beta_2$. The system composed of the concatenation of the two servers offers a minimum service curve $\beta_1 \odot \beta_2$.

Finally, to compute end-to-end delay bounds of individual traffic flows, we need the following corollary.

**Corollary 2.** (Individual flow residual service curve - Blind Multiplex)

Consider $m$ flows $f_1, f_2, ..., f_m$ of class $j$ crossing a system $n$ with the strict service $\beta^n_j$ offered to class $j$ and with $f_k$ $\alpha^{sp}_k$-constrained. The maximum packet length of $f_k$ is $l_{k,\max}$. Then the residual service curve offered to $f_k$ in a node $n$ in its path is:

$$ \beta^{n}_{j,k}(t) = (\beta^{n}_j(t) - \sum_{l \neq k} \alpha^{sp}_l(t) - \max_{l \neq k} l_{l,\max}) \uparrow $$

**B. Schedulability Conditions**

To infer the real-time guarantees of our proposed solution, we need first to define a necessary schedulability condition. The latter consists in respecting the stability condition within the network, where the maximum arrival rate of an input traffic at any crossed node has to be lower than the minimum guaranteed service rate within the node. This constraint is denoted as rate constraint. Then, we define a sufficient schedulability condition to infer the traffic schedulability, which consists in comparing the upper bound on end-to-end delay of each traffic flow to its deadline. This constraint is called deadline constraint.

For this sufficient schedulability condition, we detail the end-to-end delay expression of a flow $k$ in the class $j \in \{SCT, RC, BE\}$. $EED_{j,k}$, along its path $\Gamma$, as follows:

$$ EED_{j,k} = d^{ES}_{j,k} + d_{prop} + \sum_{sw \in \Gamma} d^{sw}_{j,k} $$

With $d^{ES}_{j,k}$ the delay within the end-system $es$ to transmit the flow $k$ of class $j$ and $d_{prop}$ the propagation delay along the path, which is generally negligible in an avionics network. The last delay $d^{sw}_{j,k}$ represents the upper bound on the delay within each switch $sw$, along the flow path, and it consists of several parts: (i) the store and forward delay at the input port, equal to $\frac{L}{\mu}$, with $L$ the length of the frame and $C$ the capacity; (ii) the technological latency due to the switching process, upper-bounded by 1μs; (iii) the output port delay due to the BLS and NP-SP scheduler. Hence, the only two unknown are the delays in the end-system and the output port of the switch. To enable the computation of upper bounds on these delays, we need to model the different parts of the network, and more particularly the BLS.

**C. System Modeling**

To compute upper bounds on end-to-end delays of different traffic classes using Network Calculus, we need to model each message flow to compute its maximum arrival curve, and the behaviour of end-systems and the crossed switches to compute the minimum service curves.

The characteristics of each traffic flow $k$ of class $j \in \{SCT, RC, BE\}$, generated by an end-system, is characterised by $\{BAG_k, MF S_k, D_{l,k}, J_k\}$ for respectively the minimum inter-arrival time, the maximum frame size integrating the protocol overhead, the deadline $D_{l,k}$ if any (generally equal to $BAG_k$ unless explicitly specified and infinite for BE) and the jitter $J_k$.

The arrival curve of traffic class $j$ in the end-system $es$, based on a leaky bucket model, is as follows:

$$ \alpha^{es}_{j}(t) = \sum_{k \in j} \alpha^{es}_{j,k}(t) = \sum_{k \in j} \frac{MF S_k}{BAG_k} (t + J_k) $$

$$ \alpha^{es}_{j}(t) = b_j + r_j \cdot t $$

For the end-systems, they are implementing a Non-Preemptive Static Priority Scheduler (NP-SP). This scheduler has been already modelled in the literature through Cor. 1 and the defined strict minimum service curve guaranteed to a traffic class $j \in \{SCT, RC, BE\}$ within an end-system $es$ is as follows:

$$ \beta^{es}_{j}(t) = \left[ C \cdot t - \sum_{k \in i, p(i) < p(j)} \alpha^{es}_{j,k}(t) - \max_{k \in i, p(i) \geq p(j)} MF S_k \right] \uparrow $$

For the proposed Extended AFDX switches, we need first to model the impact on the SCT of the BLS implemented on top of the NP-SP scheduler. To achieve this aim, we distinguish two possible scenarios. The first one covers the particular case...
where the priority of SCT remained low (2), i.e., the other queues are empty; whereas the second one covers the general case where the priority of SCT oscillates between low (2) and high (0), as explained in Section II-B. The minimum service curve guaranteed within the switch in the first scenario is due to the NP-SP scheduler and denoted $\beta_{SCT,2}^{BL}$, which is computed via Cor. 1 when considering the impact of RC traffic as the highest priority and the BE as the lowest priority. On the other hand, the minimum service curve guaranteed within the switch in the second scenario is computed via Th. 2 through the concatenation of the service curves within the BLS $\beta_{SCT}^{bls}$ (computed in Section V-A) and the NP-SP $\beta_{SCT,0}^{sp}$ (computed via Cor. 1). Therefore, we define the following relations between the service curves guaranteed within the switch $sw$ and the components $\{bls, sp\}$ for the traffic classes $\{SCT, RC\}$, $\beta_{SCT}^{sw}$ and $\beta_{RC}^{sw}$, respectively:

$$\beta_{SCT}^{sw}(t) = \max(\beta_{SCT,2}^{sp}, \beta_{SCT,0}^{sp} \otimes \beta_{SCT}^{bls}(t))$$

(2)

$$\beta_{RC}^{sw}(t) = \left[ C \cdot t - \alpha_{SCT}^{bls}(t) - \max_{k \in \{i,p\} \ge p(\text{RC})} MFS_{k} \right]_{+}$$

In Section V we will detail the minimum service curve guaranteed within the BLS $\beta_{SCT}^{bls}$ and the maximal output arrival rate $\alpha_{SCT}^{bls}$ depending on the respective maximum service curve $\gamma_{SCT}$.  

D. Computing End-to-End Delays

The computation of the end-to-end delay upper bounds follows four main steps:

(1) Computing the strict minimum service curve guaranteed to each traffic class $j \in \{SCT, RC, BE\}$ in each node $n$ of type $n \in \{es, sw\}$, $\beta_{n}^{p}$, will infer the computation of the residual service curve, guaranteed to each individual flow $k$ of class $j$, $\beta_{j,k}^{n}$ using Cor. 2.

(2) Knowing the residual service curve guaranteed to each flow within each crossed node allows the propagation of the arrival curves along the flow path, using Th 1. We can compute the output arrival curve, based on the input arrival curve and the minimum service curve, which will be in its turn the input of the next node;

(3) The computation of the minimum end-to-end service curve of each flow $k$ in class $j$, based on Th 2, is simply the concatenation of the residual service curves, $\beta_{j,k}^{n}$, $\forall n$ along its path $path_{k}$.

(4) Given the minimum end-to-end service curve of each flow $k$ in class $j$ along its $path_{k}$ and its maximum arrival curve at the initial source, the end-to-end delay upper bound $D_{j,k}$ is the maximum horizontal distance between the two curves (see Th 1).

Hence, as we can notice, we need to model all the unknown service curves, related to the BLS, to enable the end-to-end delay upper bounds computation. These curves are detailed in the next section. It is worth noting that since the BE class has no deadline, the service curves guaranteed to this class and the computation of the respective upper bounds on end-to-end delays are not detailed in this paper.

V. SERVICE CURVES USING NC

To compute the service curves offered to the burst limited traffic flows, i.e., SCT, and the non burst limited traffic flows, i.e., RC and BE, we need to detailed two types of windows, which are enforced by the BLS behaviour. The first one is denoted as sending window, during which the SCT has the highest priority and is sent until the consumed credit reaches the maximum threshold, $L_{M}$. The second one is called idle window where the SCT has the priority just higher than BE and the consumed credit is decreasing until reaching the minimum threshold, $L_{R}$. Moreover, due to the non-preemptive message transmission, both windows have minimal and maximal durations. The various combinations of such durations will induce the service curves, which are necessary for computing upper bounds on end-to-end delays and detailed in this section.

A. Strict Minimum Service Curve of SCT

The strict minimum service curve of SCT defines a lower bound on the SCT output cumulative traffic from the BLS. This curve represents the most deteriorated behaviour of BLS, in terms of offered service to the SCT, which maximizes its delay within the BLS. Hence, to cover this worst-case behaviour, we combine the maximum idle window and the minimum sending window durations.

The minimum sending window duration, $\Delta_{send}^{min}$, is the time for the consumed credit to go from the lowest to the highest thresholds, i.e., from $L_{R}$ to $L_{M}$, with an increasing slope $I_{send}$:

$$\Delta_{send}^{min} = \frac{L_{M} - L_{R}}{I_{send}}$$

(3)

The maximum idle window duration, $\Delta_{idle}^{max}$, is the time for the consumed credit to go from $L_{M}$ to $L_{R}$ with a decreasing slope $I_{idle}$, in addition to the transmission time of a maximum frame of the RC traffic. The latter is due to the non-preemption feature when a RC frame is starting its transmission just before the consumed credit reaches the lowest threshold, $L_{R}$. It is worth noting that the BE class impacts the SCT only within the NP-SP scheduler and not within the BLS since it has a priority (3) lower than the lowest priority of SCT (2):

$$\Delta_{idle}^{max} = \frac{L_{M} - L_{R} + MFS_{RC}}{I_{idle}}$$

(4)

Therefore, the strict minimum service curve guaranteed to the SCT, $\beta_{SCT}^{bls}$ is defined in Th. 2.

**Theorem 3** (Strict Minimum Service Curve of SCT in BLS). Consider a SCT crossing a server with a constant rate $C$, implementing a BLS shaper. The strict minimum service curve guaranteed to the SCT is as follows:

$$\beta_{SCT}^{bls}(t) = \frac{\Delta_{send}^{min}}{\Delta_{send}^{min} + \Delta_{idle}^{max}} C \cdot (t - \Delta_{idle}^{max})^{+}$$

(5)

where $[x]^{+}$ is the maximum between $x$ and 0.

**Proof.** Consider $R_{SCT}(t)$ the output cumulative function of the SCT at the output of the server implementing a BLS, and $\Delta R_{SCT}(\delta)$ the variation of the output cumulative function
during δ. To prove that \( \beta_{sct}^{bls} \) in Eq. (5) is a strict minimum service curve, we need to prove Def. 2 for any backlogged period δ, i.e., the SCT flows are continuously backlogged during δ.

During a backlogged period δ, the SCT has at least p opportunities of full service constrained by \( \beta(t) = C \cdot t \) during the minimum sending window \( \Delta_{send}^{min} \), then:

\[
\Delta R_{SCT}(\delta) \geq C \cdot \Delta_{send}^{min}
\]

(6)

The main idea is to find a lower bound of p to define the service curve guaranteed to SCT, \( \beta_{sct}^{bls} \). On the other hand, if SCT has p opportunities to be transmitted, then the RC traffic (since it is the only traffic class with a priority higher than the lowest priority of SCT during the idle window) has at most \((p+1)\) opportunities to be transmitted during at the worst-case the maximum idle window, \( \Delta_{idle}^{max} \), then:

\[
\Delta R_{RC}(\delta) \leq (p+1) \cdot C \cdot \Delta_{idle}^{max}
\]

(7)

Giving the strict service curve property of \( C \cdot t \) since we have a constant rate server and using Eq. (7), we have:

\[
C \cdot \delta \leq \Delta R_{SCT}^*(\delta) + \Delta R_{RC}^*(\delta)
\]

\[
\leq \Delta R_{SCT}^*(\delta) + (p+1) \cdot C \cdot \Delta_{idle}^{max}
\]

Consequently, the lower bound of p is as follows:

\[
p \geq \frac{C \cdot \delta - \Delta R_{SCT}^*(\delta)}{C \cdot \Delta_{idle}^{max}} - 1
\]

(8)

When injecting Eq. (8) in Eq. (6), we obtain:

\[
\Delta R_{SCT}^*(\delta) \geq \left( \frac{C \cdot \delta - \Delta R_{SCT}^*(\delta)}{C \cdot \Delta_{idle}^{max}} - 1 \right) \cdot C \cdot \Delta_{send}^{min}
\]

\[
\Delta R_{SCT}^*(\delta) \cdot \left( 1 + \frac{\Delta_{send}^{min}}{\Delta_{idle}^{max}} \right) \geq \left( \frac{\delta}{\Delta_{idle}^{max}} - 1 \right) \cdot C \cdot \Delta_{send}^{min}
\]

\[
\Delta R_{SCT}^*(\delta) \geq \frac{\Delta_{idle}^{max} - 1}{\Delta_{idle}^{max}} \cdot C \cdot \Delta_{send}^{min}
\]

Giving that \( \Delta R_{SCT}^*(\delta) \geq 0 \), then:

\[
\Delta R_{SCT}^*(\delta) \geq \frac{\Delta_{send}^{min}}{\Delta_{send}^{min} + \Delta_{idle}^{max}} \cdot C \cdot (\delta - \Delta_{idle}^{max})^+
\]

B. Maximum Service Curve of SCT

The maximum service curve of SCT represents the best offered service to the SCT, which induces the minimum processing delay within the BLS. As such, in the presence of RC traffic, we combine the minimum idle window duration and the maximum sending window one to handle this best-case behaviour.

The maximum sending window duration, \( \Delta_{send}^{max} \), is equal to the sum of: (i) the minimum sending window duration, \( \Delta_{send}^{min} \); (ii) the transmission time of a maximum frame of the SCT due to the non-preemption feature, i.e., one SCT frame may start its transmission just before the consumed credit reaches \( L_M \); (iii) the time to consume the gained credit during the transmission of one additional maximum frame of RC traffic at the end of the idle window. The latter parameter is due to the fact that the resume level of BLS, \( L_R \), is the lower threshold on the consumed credit to trigger the priority change of the SCT from lowest to highest, and not an extreme value for the consumed credit itself. Actually, if a frame of RC traffic has been transmitted just at the end of the idle window, the consumed credit keeps decreasing until it either reaches 0, or the transmission ends. Therefore, the lowest value the consumed credit can reach due to the non-preemption feature is \( max(0, L_R - \frac{MFS_{SCT}}{C} \cdot \Delta_{idle}) \). The additional time during which the consumed credit can then increase with a slope of \( I_{send} \) is \( L_R - \frac{MFS_{SCT}}{C} \cdot \Delta_{send} \). The maximum sending window duration is then as follows:

\[
\Delta_{send}^{max} = \frac{L_M - L_R}{I_{send}} + \frac{MFS_{SCT}}{C} + \min \left( \frac{MFS_{RC}}{C} \cdot I_{idle}, \frac{L_R}{I_{send}} \right)
\]

(9)

However, it is worth noting that the consumed credit may start at 0, such as at the initialisation phase or after a long period of inactivity. Hence, the maximum sending window duration covering such possibility, \( \Delta_{send,0}^{max} \), is as follows:

\[
\Delta_{send,0}^{max} = \frac{L_M - L_R}{I_{send}} + \frac{MFS_{SCT}}{C}
\]

(10)

The minimum idle window duration, \( \Delta_{idle}^{min} \), is simply the time it takes for the consumed credit to go from \( L_M \) to \( L_R \) with a decreasing slope of \( I_{idle} \):

\[
\Delta_{idle}^{min} = \frac{L_M - L_R}{I_{idle}}
\]

(11)

Therefore, the maximum service curve guaranteed to the SCT, \( \gamma_{SCT}^{bls} \), is defined in Th. 4.

**Theorem 4** (Maximum Service Curve of SCT in BLS). Consider a SCT crossing a server with a constant rate \( C \), implementing a BLS shaper. The maximum service curve guaranteed to the SCT is as follows:

\[
\gamma_{SCT}^{bls}(t) = \begin{cases} 
C \cdot t & \text{if no RC traffic: } C \cdot t \\
\frac{\Delta_{send}^{max}}{\Delta_{send}^{max} + \Delta_{idle}^{max}} \cdot C \cdot t + \Delta_{send}^{max} \cdot \Delta_{idle}^{min} & \text{Otherwise}
\end{cases}
\]

with \( \Delta_{nom}^{SCT} = \Delta_{send}^{max} + \Delta_{idle}^{min} \).

**Proof.** First, it is obvious that in the absence of RC traffic, SCT can use the maximum service \( \gamma(t) = C \cdot t \). Then, for the more general case, consider \( R_{SCT}(t) \) the output cumulative function of the SCT at the output of the server implementing a BLS, and \( \Delta R_{SCT}^*(\delta) \) the variation of the output cumulative function during \( \delta \).

During a backlogged period \( \delta \) for SCT and RC traffic, the SCT has at most \( p+1 \) opportunities with \( p \) times a full service constrained by \( \gamma \) during the maximum sending window \( \Delta_{send}^{max} \), in addition to once during \( \Delta_{send}^{min} \), then:

\[
\Delta R_{SCT}^*(\delta) \leq p \cdot C \cdot \Delta_{send}^{max} + C \cdot \Delta_{send}^{min}
\]

(12)

The main idea is to find an upper bound of \( p \) to define the maximum service curve guaranteed to SCT, \( \gamma_{SCT}^{bls} \). On the
other hand, if SCT has at most \( p + 1 \) opportunities to be transmitted, then the RC traffic has at least \( p \) opportunities to be transmitted during the minimum idle window, \( \Delta_{\text{idle}}^{\text{min}} \), then:

\[
\Delta R_{RC}^*(\delta) \geq p \cdot C \cdot \Delta_{\text{idle}}^{\text{min}} \quad (13)
\]

Giving the maximum service curve property of \( \gamma \) and using Eq. (13), we have:

\[
C \cdot \delta \geq \Delta R^*_{SCT}(\delta) + \Delta R^*_{RC}(\delta) \geq \Delta R^*_{SCT}(\delta) + p \cdot C \cdot \Delta_{\text{idle}}^{\text{min}}
\]

Consequently, the upper bound of \( p \) is as follows:

\[
p \leq \frac{C \cdot \delta - \Delta R^*_{SCT}(\delta)}{C \cdot \Delta_{\text{idle}}^{\text{min}}}
\]

(14)

When injecting Eq. (14) in Eq. (12), we obtain:

\[
\Delta R^*_{SCT}(\delta) \leq \frac{C \cdot \delta - \Delta R^*_{SCT}(\delta)}{C \cdot \Delta_{\text{idle}}^{\text{min}}} \cdot C \cdot \Delta_{\text{send}}^{\text{max}} + C \cdot \Delta_{\text{send},0}^{\text{max}}
\]

(15)

\[
\Delta R^*_{SCT}(\delta) \cdot \left(1 + \frac{\Delta_{\text{send}}^{\text{max}}}{\Delta_{\text{idle}}^{\text{min}}} \right) \leq \frac{\delta}{\Delta_{\text{idle}}^{\text{min}}} \cdot C \cdot \Delta_{\text{send}}^{\text{max}} + C \cdot \Delta_{\text{send},0}^{\text{max}}
\]

(16)

\[
\Delta R^*_{SCT}(\delta) \leq \frac{\Delta_{\text{nom}}^{\text{idle}}}{\Delta_{\text{nom}}^{\text{idle}}} \cdot C \cdot \Delta_{\text{send}}^{\text{max}} + C \cdot \Delta_{\text{send},0}^{\text{max}}
\]

\[
\Delta_{\text{nom}}^{\text{idle}} = \Delta_{\text{send}}^{\text{max}} + \Delta_{\text{idle}}^{\text{min}}.
\]

C. Maximum Output Arrival Curve of SCT

The maximum output arrival curve of SCT, from the BLS is detailed in the following Corollary:

Corollary 3 (Maximum Output Arrival Curve of SCT from BLS). Consider a SCT with a maximum leaky-bucket arrival curve \( \alpha \) at the input of a BLS shaper, guaranteeing a minimum rate-latency service curve \( \beta_{SCT}^{\text{bls}} \) and a maximum service curve \( \gamma_{SCT}^{\text{bls}} \). The maximum output arrival curve is:

\[
\alpha_{SCT}^{*,\text{bls}}(t) = \min(\gamma_{SCT}^{\text{bls}}(t), \alpha \odot \beta_{SCT}^{\text{bls}}(t))
\]

(15)

Proof. To prove Corollary 3 we generalize herein the rule 13 in p. 123 in [8], i.e., \((f \odot g) \odot h \leq f \odot (g \odot h)\), to the case of three functions \( f, g, h \) when \( g \odot h \in \mathcal{F} \), where \( \mathcal{F} \) is the set of non negative and wide sense increasing functions:

\[
\mathcal{F} = \{ f : \mathbb{R}^+ \to \mathbb{R}^+ \mid f(0) = 0, \forall t \geq s : f(t) \geq f(s) \}
\]

According to Theorem 1 we have \( \alpha^*(t) = (\gamma_{SCT}^{\text{bls}} \odot \alpha) \odot \beta_{SCT}^{\text{bls}} \). Moreover, in the particular case of a leaky-bucket arrival curve \( \alpha \) and a rate-latency service curve \( \beta_{SCT}^{\text{bls}} \), \( \alpha \odot \beta_{SCT}^{\text{bls}} \) is a leaky-bucket curve, which is in \( \mathcal{F} \). Hence, we have the necessary condition to prove the following:

\[
(\alpha \odot \gamma) \odot \beta(t) \leq \gamma \odot (\alpha \odot \beta)(t) \leq \min(\gamma(t), \alpha \odot \beta(t))
\]

D. Minimum Strict Service Curves of RC

Theorem 5 (Minimum Strict Service Curves of RC). Consider a SCT with a maximum leaky-bucket arrival curve \( \alpha \) at the input of a server with a constant rate \( C \) implementing a BLS shaper, guaranteeing a minimum rate-latency service curve \( \beta_{SCT}^{\text{bls}} \) and a maximum service curve \( \gamma_{SCT}^{\text{bls}} \). The server is crossed by SCT and RC traffic, where SCT has the highest priority. The minimum strict service curve guaranteed to RC traffic in the NP-SP, integrating the impact of the BLS, is as follows:

\[
\beta_{RC}^{*,\text{bls}}(t) = \left[ \max(\beta_{RC}^{*,\text{bls}}(t), \beta_{RC}^{\text{bls}}(t)) - \max_{k \in i(p(i) \geq p(RC))} MFS_k \right] \uparrow
\]

where:

\[
\beta_{RC}^{*,\text{bls}}(t) = \left( C \cdot t - \alpha_{SCT}^{*,\text{bls}}(t) \right) \odot \beta_{SCT}^{\text{bls}}(t)
\]

(17)

Proof. According to Cor. 1 the residual minimum strict service curve guaranteed to RC traffic crossing a NP-SP scheduler is as follows:

\[
\beta_{RC}^*(t) = \left[ C \cdot t - \alpha_{SCT}^{*,\text{bls}}(t) - \max_{k \in i(p(i) \geq p(RC))} MFS_k \right] \uparrow
\]

Moreover, according to Cor. 3 the maximum output arrival curve of SCT from the BLS, \( \alpha_{SCT}^{*,\text{bls}}(t) \), is:

\[
\alpha_{SCT}^{*,\text{bls}}(t) = \min(\gamma_{SCT}^{\text{bls}}(t), \alpha \odot \beta_{SCT}^{\text{bls}}(t))
\]

(18)

VI. PERFORMANCE ANALYSIS

In this section, we conduct performance analysis of the proposed Extended AFDX to evaluate its efficiency to support mixed criticality data, in comparison to the legacy AFDX solution (i.e. AFDX with regular NP-SP scheduler). This evaluation is based on the worst-case timing analysis methodology and the various service curves detailed in Sections V and VI respectively. First, we describe our realistic avionics case study. Afterwards, we assess the scalability of the Extended AFDX in comparison to the legacy one based on the current avionics traffic configuration. Finally, we analyse the impact
of our proposal on SCT and RC performance, in comparison to the legacy solution, when considering future avionics traffic configurations.

A. Avionics Case Study

Our case study is a representative avionics communication architecture of the A380, based on a 1-Gigabit AFDX backbone network, which consists of 4 switches and 64 end-systems as shown in Fig. 3(a). The maximum utilisation rate on the 100Mbps legacy AFDX on board the A380 is 30%. Thus, on the 1Gigabit version, the maximum utilisation rate will be only of 3%. However, there is only legacy AFDX traffic, i.e., RC, circulating on this current communication architecture. Hence, to enable the performance analysis of our proposed Extended AFDX, we have extended this current traffic configuration, denoted herein as legacy reference configuration, to support different traffic profiles generated by each end-system, which are described in Table III. Each traffic class is characterised by (MFS, BAG, DI, J) as detailed in Section IV-C. The SCT has a deadline of 2ms, and because of the BLS behaviour it admits the highest priority 0 during the sending windows and the priority below RC (2) during the idle windows. Each circulating traffic flow on the backbone network is a multicast flow with 16 destinations, and crosses two successive switches before reaching its final destinations. The first switch in the path receives traffic from 16 end-systems to forward it in a multicast way to its two neighbouring switches. Afterwards, the second switch in the path, which receives traffic from the two predecessor switches, forwards the traffic in its turn to the final end-system. Figure 3(b) shows the traffic communication patterns between the source and the final destinations of a given flow.

![Traffic communication patterns between the source and the final destinations of a given flow.](image)

The main considered performance metrics are: (i) the maximum utilisation rate of each traffic class, that can be sent on the Extended AFDX architecture while respecting the system constraints described in Section IV-B. This metric enables the scalability analysis of our proposal, in comparison with the legacy AFDX; (ii) the delay bound of SCT and RC classes to prove the predictability of the Extended AFDX and analyse its impact on the system timing performance, in comparison with the legacy AFDX. It is worth noting that because the BE does not have a deadline, and its largest impact on the other priorities is the transmission time of a maximum sized frame, the timing performance of this class is not detailed in this paper.

To compute both performance metrics, we consider four scenarios, i.e., scenarios 1 and 2 are for the scalability analysis; whereas scenarios 3 and 4 are for timing analysis.

Concerning scenarios 1 and 2, we have started from the legacy reference configuration, and then computed the maximum utilisation rate of SCT class that can be transmitted on the legacy AFDX, while respecting the system constraints in the presence of 3% of RC traffic. This computation has shown that the legacy AFDX can support up to a maximum utilisation rate for SCT of $UR_{SCT}^m = 28.7\%$. Hence, scenario 1 (resp. scenario 2) consists in starting from a traffic configuration characterised by ($UR_{SCT}^m = 28.7\%; UR_{RC}^m = 3\%$) circulating on the Extended AFDX, then increasing the $UR_{SCT}^m$ (resp. $UR_{RC}^m$) until finding the maximum value which still respects the system constraints. The aim of this scenario is to compare the scalability of the Extended and legacy AFDX solutions, when increasing the congestion due to SCT (resp. RC) traffic. The results of both scenarios (1 and 2) are detailed in Section VI-B.

Afterwards, to have an idea about the timing performance of future avionics configurations based on the 1 Gigabit AFDX technology, which may very probably support higher utilisation rate of SCT and RC traffic than the legacy reference configuration, we consider scenarios 3 and 4 described in Table IV. As it can be noticed, the principle of scenario 3 (resp. scenario 4) is to fix the utilisation rate of RC class (resp. SCT class) at 20% and vary the SCT (resp. RC) utilisation rate to assess the impact of increasing network congestion on the timing performance. The considered BLS parameters are the same for both scenarios: $BW = 0.46$ to support a maximum utilisation rate of SCT of $UR_{SCT}^m = 45\%$ (this is an upper bound for the estimated future needs in terms of SCT traffic); $L_R = 0$ and $L_M = 22,118$ to enable the transmission of a maximum SCT burst within the BLS of 80 frames during a minimum sending window, i.e., a generated burst of 5 SCT flows per End-System. Moreover, as it is illustrated in Table IV, the variation of the utilisation rate of a class $i$ is obtained through increasing the number of generated traffic flows within each end-system, $n_i^c$. Thus, the bottleneck utilisation rate is equal to $UR_{i}^c = \frac{n_i^c}{C_i}$ with $C_i$ the capacity used in the bottleneck by the aggregated flows of class $i \in \{RC, SCT\}$ and $C_i = 16 \cdot n_i^c \cdot \frac{MFS\_BM}{\text{BAG\_ES}}$. The results of both scenarios (3 and 4) are detailed in Section VI-C.

![Representative AFDX network: (a) Architecture; (b) Traffic communication patterns](image)
B. Analysing the scalability of the current avionics configuration

| Configurations | Legacy reference | Scenario 1 | Scenario 2 |
|----------------|------------------|------------|------------|
| \( UR_{SCT}^{bn} \) (%) | 28.7 | 43 | 28.7 |
| \( UR_{RC}^{bn} \) (%) | 3 | 3 | 13 |
| \( (L_M; L_R) \) | N.A | (10, 240: 0) | (35, 840: 0) |
| BW | N.A | 0.90 | 0.65 |

TABLE V
RESULTS OF SCENARIOS 1 AND 2

The aim of this section is to analyse the scalability of our proposed Extended AFDX, in comparison with the legacy AFDX. Hence, from the legacy reference configuration characterised by \( UR_{RC}^{bn} = 3\% \), \( UR_{SCT}^{bn} = 28.7\% \), we have tested for scenario 1 (resp. scenario 2) various BLS parameters to increase as much as possible the maximum SCT (resp. RC) utilisation rate. As shown in Table V there exists a BLS configuration for scenario 1 (resp. scenario 2) allowing to achieve an utilisation rate of SCT (resp. RC) of \( UR_{SCT}^{bn} = 43\% \) (resp. of \( UR_{RC}^{bn} = 13\% \)) under Extended AFDX, instead of only \( UR_{SCT}^{bn} = 28.7\% \) (\( UR_{RC}^{bn} = 3\% \)) under legacy AFDX.

These results show an enhancement of scalability with the Extended AFDX of 50\% and 333\% for the SCT and RC classes, respectively, in comparison with the legacy AFDX.

C. Analysing timing performance of future avionics configurations

SCT timing performance

To analyse the timing performance of SCT when using the Extended AFDX instead of the legacy AFDX, we focus on Figures (a) and (b) showing the SCT delay bounds evolution, regarding the SCT and RC bottleneck utilisation rate variation, respectively.

As shown in Fig. (a), when increasing the bottleneck utilisation rate of the SCT, the delay upper bounds are obviously increasing under both solutions, but are globally higher under the Extended AFDX. This fact is due to the BLS behaviour on top of the NP-SP scheduler implemented within the Extended AFDX, which infers dividing the SCT burst to be sent within many sending windows; whereas the regular NP-SP scheduler implemented within the legacy AFDX is sending the SCT burst all at once.

On the other hand, as it can be noticed in Fig. (a), when increasing the bottleneck utilisation rate of the RC, the SCT delay bounds are constant under the legacy AFDX since SCT has the highest priority level and is at most delayed by a maximum sized frame of lower priorities, i.e., RC and BE; whereas they are increasing under the Extended AFDX for a RC utilisation rate up to 20\% and become equal to the SCT deadline (2ms) for a RC utilisation rate higher than 20\%. The increase is due to the fact that the RC rate is not large enough to use all the bandwidth guaranteed by the BLS; thus the guaranteed SCT service within the BLS is limited by the left part of the Eq. (2). This shows the good isolation level, enforced by the BLS, between the mixed-criticality traffic, i.e., RC and SCT.

These results show the impact of the Extended AFDX network on SCT when increasing the network congestion. The main interesting feature to highlight is its efficiency to guarantee a high isolation level between the mixed criticality traffic, which is one of the key requirements for avionics applications.

RC timing performance

We detail herein the main interesting results concerning the impact of our proposed solution on the RC timing performance, based on Figures (b) and (b).

Fig. (b) illustrates the variation of the RC delay upper bounds in terms of the SCT bottleneck utilisation rate. We can easily distinguish two phases on this figure. The first one is observed for a maximum utilisation rate below 14\%, where the delay upper bounds under both solutions are very similar. The second phase, i.e., when the maximum utilisation rate is higher than 14\%, shows that the delay upper bounds increase inherently under legacy AFDX, whereas they are constant under the Extended AFDX.

These results are coherent with the guaranteed service to the RC traffic in Th. which is the maximum between \( \beta_{RC}^{bn} \) and
first one concerns the noticeable RC delay bounds decrease, where they become constant after a given SCT utilisation rate; (ii) the second one is the enhancement of the RC delay bound under the Extended AFDX when varying the RC utilisation rate, which is up to 74%.

VII. CONCLUSIONS

In this paper, we have defined and analysed the timing performance of an Extended AFDX network, to handle mixed criticality avionics applications. The proposed solution implements a BLS shaper on top of NP-SP within AFDX switches, to manage three priority levels. The conducted performance analysis on a realistic avionics case study highlights the benefit of using such a proposal, to isolate the highest priority and mitigate the impact of highest priority traffic on the RC one. Numerical results have shown noticeable enhancements of the delay upper bounds of the RC traffic (up to 74%), and a gain in terms of maximum utilisation rate up to 333% for RC and 50% for SCT, in comparison with the legacy AFDX network.

As a next step, we will introduce a tuning method to find the best BLS parameters, which respect the highest priority traffic deadline, while decreasing as much as possible the RC delay bounds.

Fig. 5. Scenario 4: Impact of RC max. utilisation rate on: (a) SCT delays; (b) RC delays

\(\beta_{\text{RL}}\). Hence, during the first phase, the service corresponds to \(\beta_{\text{RL}}^{\text{up}}\), which is impacted by the maximum arrival curve of the SCT; thus its maximum utilisation rate. This fact explains the delay bounds increase. Afterwards, the service becomes related to \(\beta_{\text{RL}}^{\text{hys}}\) during the second phase, which enforces a maximum constraint on the arrival curve of the SCT under the Extended AFDX due to the BLS, \(\beta_{\text{RL}}^{\text{hys}}/\beta_{\text{SCT}}\). This maximum constraint implies a constant delay under the Extended AFDX.

On the other hand, the service guaranteed under legacy AFDX is deeply related to the arrival curve of SCT, which explains the inherent delay bound increase.

Fig. 5 (b) shows the impact of the RC utilisation rate variation on the RC delay bounds. As it can be noticed, the RC delay bounds are increasing under both solutions, but still are better under Extended AFDX. For instance, for a RC utilisation rate of 10%, we observe a delay bound of 1.5ms and 0.9ms under the legacy and Extended AFDX, respectively; thus the enhancement of the delay bound is about 40% at \(U_{\text{RL}}^{\text{up}} = 10\%\), and it goes up to 74% at \(U_{\text{RL}}^{\text{hys}} = 2\%\).

We need also to highlight the scalability enhancement in terms of utilisation rate of SCT and RC traffic under the Extended AFDX, in comparison to the legacy AFDX, which is coherent with the conclusions of the Section VI-B. In scenario 3 (resp. scenario 4), the maximum SCT (resp. RC) utilisation rate, respecting all the constraints, is 19.5% (resp. 22.5%) and 18.5% (resp. 18.5%) with Extended and legacy AFDX, respectively.

These results show the valuable impact of the Extended AFDX on RC traffic, in comparison with the legacy AFDX solution. We can distinguish two interesting features: (i) the

\[\begin{array}{ll}
\text{REFERENCES} \\
[1] Airlines Electronic Engineering Committee. Aircraft Data Network Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network, ARINC Specification 664. Aeronautical Radio, 2002. \\
[2] Anne Bouillard, Laurent Jouhet, and Eric Thierry. Service curves in Network Calculus: dos and don’ts. Research report, INRIA, 2009. \\
[3] Rodrigo Coelho, Gerhard Fohler, and J-L. Scharbarg. Worst-case backlog for AFDX network with n-priorities. In RTN, 2014. \\
[4] Markus Fidler, Volker Sander, and Wojciech Klimala. Traffic shaping in aggregate-based networks: implementation and analysis. Computer Communications, 2005. \\
[5] Franz-Josef Gotz. Traffic Shaper for Control Data Traffic (CDT). IEEE 802 AVB Meeting. \\
[6] Jérôme Grieu. Analyse et évaluation de techniques de commutation Ethernet pour l’interconnexion des systèmes avioniques. PhD thesis, INPT, 2004. \\
[7] S.-K. Kweon and K. Shin. Achieving real-time communication over Ethernet with adaptive traffic smoothing. In RTAS, 2000. \\
[8] J.Y. Le Boudec and P. Thiran. Network calculus: a theory of deterministic queuing systems for the internet. Springer-Verlag, 2001. \\
[9] Jork Loeser and Hermann Haertig. Low-latency hard real-time communication over switched ethernet. In ECRTS, IEEE, 2004. \\
[10] Zhao Luxi, Pop Paul, Li Qiao, Chen Junyan, and Xiong Huagang. Timing analysis of rate-constrained traffic in TTEthernet using network calculus. Real-Time Systems, 2017. \\
[11] Ahlem Mifdouai, Fabrice Frances, and Christian Fraboul. Full Duplex Switched Ethernet for Next Generation "1553B"-Based Applications. In RTAS, 2006. \\
[12] R. Bosch GmbH. CAN specification Version 2.0. Technical report, 1991. \\
[13] E.-J. Gotz, S. Kerschbaum, and F. Chen. Towards the Calculation of Performance Guarantees for BLS in Time-Sensitive Networks. IEEE 802.1 TSN Meeting, 2013. \\
[14] N. Concer S. Thangamuthu, P. Cuijpers JL, and J. Lukkien. Analysis of ethernet-switch traffic shapers for in-vehicle networking applications. In Design, Automation & Test in Europe Conference & Exhibition, IEEE, 2015. \\
[15] Perathoner Simon, Wandeler Ernesto, Thiele Lothar, and et al. Influence of different abstractions on the performance analysis of distributed hard real-time systems. Design Automation for Embedded Systems, 2009. \\
[16] Standard. IEEE Std. 802.1Qav, IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams, 2009.
\end{array}\]
[17] W. Steiner, G. Bauer, B. Hall, M. Paulitsch, and S. Varadarajan. TTEthernet Dataflow Concept. In Eighth IEEE International Symposium on Network Computing and Applications, 2009.

[18] W. Steiner, P. Heise, and S. Schneele. Recent ieee 802 developments and their relevance for the avionics industry. In 2014 IEEE/AIAA 33rd DASC.

[19] Michael Johas Teener. Back to the future:using TAS and preemption for deterministic distributed delays. IEEE 802.1 AVB TG Meeting, San Antonio.

[20] D. Thiele and R. Ernst. Formal worst-case timing analysis of Ethernet TSN’s burst-limiting shaper. In DATE, 2016.

[21] Daniel Thiele, Rolf Ernst, and Jonas Diemer. Formal worst-case timing analysis of Ethernet TSN’s time-aware and peristaltic shapers. In VNC. IEEE, 2015.

[22] TSN Task Group. TSN Specifications. [http://www.ieee802.org/1/pages/tsn.html](http://www.ieee802.org/1/pages/tsn.html)

[23] Tianran ZHOU and Xiong Huagang. Design of energy-efficient hierarchical scheduling for integrated modular avionics systems. Chinese Journal of Aeronautics, 2012.