Performance Enhancement of Extended AFDX via Bandwidth Reservation for TSN/BLS Shapers

A. FINZI, A. MIFDAOUI, F. FRANCES and E. LOCHIN
University of Toulouse-ISAE, France

ABSTRACT

To support mixed-criticality applications, the AFDX may integrate multiple traffic classes: Safety-Critical Traffic (SCT) with hard real-time constraints, Rate-Constrained (RC) traffic requiring bounded latencies and Best Effort (BE) traffic with no delivery constraints. These traffic classes are managed based on a Non-Preemptive Strict Priority (NP-SP) Scheduler, where the highest priority traffic (SCT) is shaped with a Burst Limiting Shaper (BLS). The latter has been defined by the Time Sensitive Networking (TSN) task group to limit the impact of high priority flows on lower priority ones. This paper proposes two bandwidth reservation methods for BLS shapers in AFDX networks. The proposed methods are evaluated on a realistic AFDX configuration. Results show their efficiency to noticeably enhance the RC delay bounds and the SCT schedulability, in comparison to an intuitive method.

ACM Reference Format:

1 INTRODUCTION

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 advantages, such as easier installation and maintenance and reduced weight and costs. This homogeneous communication architecture, based on the AFDX technology, needs to support mixed-criticality applications, where safety-critical and best effort traffic co-exist. Hence, in addition to the current AFDX traffic profile, called Rate Constrained (RC) traffic, at least two extra profiles have to be handled. The first, denoted by Safety-Critical Traffic (SCT), is specified to support flows with hard real-time constraints and the highest criticality, e.g., flight control data; whereas the second is for Best-Effort (BE) flows with no delivery constraint and the lowest criticality, e.g., In-Flight Entertainment traffic.

To cope with this emerging issue, in [4], we have assessed the most relevant existing solutions enabling mixed-criticality on the AFDX vs avionics requirements. The Burst-Limiting Shaper (BLS) [5] (defined in the Time Sensitive Networking (TSN) task group [6]) on top of Non-Preemptive Strict-Priority (NP-SP) scheduler has been selected as the most promising solution favoring the main avionics requirements. In particular, the fairness of such a solution has been highlighted compared to the current AFDX implementing NP-SP. Preliminary performance evaluation of such a solution, denoted extended AFDX, has been provided based on simulations. The first results were encouraging to pursue this line through providing formal timing analysis to prove certification requirements, a key point in avionics. In particular, simulations showed the ability of the BLS to limit the impact of SCT on RC delay bounds (up to 40% delay bound decrease) and enhance schedulability up to 48%. Afterwards, in [3], we have introduced a Network Calculus-based approach to compute the delay bounds of SCT and RC classes in such an extended AFDX network, taking into account the impact of the TSN/BLS. The performance evaluation of our proposal on a realistic AFDX configuration has highlighted its efficiency, in comparison with the current AFDX (implementing only NP-SP scheduler).

In this paper, our aim is to provide optimized bandwidth reservation methods with different complexity levels for the TSN/BLS shapers, to enhance as much as possible the schedulability and the delay bounds of the different traffic classes. These methods are based on an extension of the timing analysis introduced in [3] to cover multi-hop communication. We perform a set of experiments to assess the efficiency of our introduced methods, with reference to an intuitive bandwidth reservation method. Similar works have been conducted to define bandwidth reservation methods for Credit Based Shaper (CBS) of AVB networks [1, 2]. However, to the best of our knowledge, this issue has not been handled yet in the literature for TSN/BLS shapers in avionics domain.

The rest of the paper is organized as follows. Section II introduces the system model. Section III details two optimized bandwidth reservation methods with different complexity levels for TSN/BLS shapers. Section IV evaluates the proposed methods on a realistic avionic configuration to assess their efficiency. Finally, Section VI concludes the paper.

2 SYSTEM MODEL AND ASSUMPTIONS

In this section, we first describe the extended AFDX switch model. Then, we detail the BLS behavior and its main parameters. Finally, we present the considered schedulability condition and traffic model. The main notations are presented in Table 1.

2.1 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 SCT with its priority set by the BLS and the tightest temporal deadline, e.g., Flight-control flows; (ii) the RC with the medium priority and a deadline constraint...
to guarantee, e.g., current AFDX flows; (iii) the BE with the lowest priority and no time constraint, e.g., In-Flight Entertainment.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Link speed</td>
</tr>
<tr>
<td>MFS_f</td>
<td>Maximum Frame Size of flow f</td>
</tr>
<tr>
<td>BAG_f</td>
<td>Bandwidth Allocation Gap of flow f</td>
</tr>
<tr>
<td>J_f, D_f</td>
<td>Jitter and deadline of flow f</td>
</tr>
<tr>
<td>L_M, L_R, BW</td>
<td>BLS maximum and resume credit levels, BLS reserved bandwidth</td>
</tr>
<tr>
<td>I_idle, I_end</td>
<td>BLS idle and sending slopes</td>
</tr>
<tr>
<td>p(j)</td>
<td>Priority level of a class j with p(j) ∈ {0, 1, 2, 3}</td>
</tr>
<tr>
<td>UR_i^Bls</td>
<td>bottleneck network utilisation rate of a class j</td>
</tr>
<tr>
<td>EDD_{j,f}</td>
<td>end-to-end delay of flow f of class j</td>
</tr>
<tr>
<td>r_f, b_f</td>
<td>rate and burst of flow f</td>
</tr>
<tr>
<td>R^mus</td>
<td>minimum guaranteed rate of class j in max</td>
</tr>
</tbody>
</table>

**Table 1: Notations**

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.

**Figure 1: An extended AFDX switch architecture**

### 2.2 BLS Shaper

The BLS belongs to the credit-based shapers class and it is generally used on top of Non-Preemptive Static Priority (NP-SP) scheduler. It has been defined in [5] by an upper threshold L_M, a lower threshold L_R, such as 0 ≤ L_R < L_M, and a reserved bandwidth BW. Additionally, the BLS is characterized by: (i) a decreasing rate I_idle = BW · C, where C is the link speed and BW is the percentage of bandwidth reserved for BLS frames; (ii) an increasing rate I_end = C − I_idle. Finally, priority of a queue q shaped by BLS, denoted p(q), can vary between a high and a low value (with 0 the highest), denoted p_H and p_L.

The behavior of the BLS is illustrated in Fig. 2 for two arrival scenarios. As shown, the credit is always between 0 and L_M and varies as follows:

(i) initially, the credit counter starts at 0 and the priority of the shaped queue is high;

(ii) the main feature of the BLS is the change of priority p(k) of the shaped queue, which occurs in two contexts: 1) if p(k) is high and credit reaches L_M; 2) if p(k) is low and credit reaches L_R;

(iii) when a frame is transmitted, the credit increases (is consumed) with a rate of I_end, 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 0 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.

**Figure 2: BLS credit evolution**

#### 2.3 Schedulability Condition and Traffic Model

First, we define a sufficient schedulability condition for SCT and RC classes, which consists in verifying that the end-to-end delay bound of each traffic flow is lower than its deadline.

The end-to-end delay expression of a flow f in the class j ∈ {SCT, RC}, EDD_{j,f}, along its path path_f is as follows:

\[
EDD_{j,f} = d^{fs}_{j,f} + d_{prop} + \sum_{i \in path_f} d_{sw,i}^{f}\quad (1)
\]

With \(d^{fs}_{j,f}\) the delay within the end-system (es) to transmit the aggregate traffic of class j and \(d_{prop}\), the propagation delay along the path, which is generally negligible in an avionics network. The last delay \(d_{sw,i}^{f}\) represents the delay within the ith switch (sw) along the flow path and it consists of several parts: (i) the store and forward delay at the input port, equal to the transmission time of a maximum-sized frame; (ii) the technological latency due to the switching process, upper-bounded by 1µs; (iii) the delay of the output port multiplexer (mux) due to the BLS (bls) and NP-SP (sp) scheduler. To enable the computation of upper bounds on these delays, the different parts of the network, and more particularly the BLS have been modeled in terms of service curves [3], where \(p^Bls\) is the service guaranteed for traffic class j in node n ∈ {es, sw} or a component n ∈ {bls, sp}.

Secondly, each traffic flow f of class j ∈ {SCT, RC, BE}, generated by an end-system, is characterized by \(\text{BAG}_j, \text{MFS}_j, \text{D}_j, \text{J}_j\) for respectively the minimum inter-arrival time, the maximum frame size integrating the protocol overhead, the deadline if any (generally equal to \(\text{BAG}_j\) unless explicitly specified and infinite for BE) and the jitter.

The arrival curve of each flow f in class j at the input of the ith node n ∈ {es, sw} or a component n ∈ {bls, sp} along its path is modeled as a leaky-bucket curve with a burst \(b^{n,i}_{j,f}\) and a rate \(r_f\):

\[
a^{n,i}_{j,f}(t) = b^{n,i}_{j,f} + r_f \cdot t
\]

For instance, the input arrival curve of flow f in class j at the end-system is: \(a^{n,i}_{j,f}(t) = MFS + \frac{MFS}{\beta_{j,f}} \cdot (t + J_f)\).

Therefore, the arrival curve of the aggregate traffic in class j at the input of the ith node n ∈ {es, sw} or a component n ∈ {bls, sp}
is: \( a_j^{n,i}(t) = \sum_{f \in j} a_j^{n,i} f(t) \). For instance,

\[
a_j^{C}(t) = b_j + r_j t \quad \text{with} \quad \begin{cases} b_j &= \sum_{f \in j} MFS_f + \frac{MFS_f}{\text{BAG}_f} J_f \\ r_j &= \sum_{f \in j} \frac{\text{BAG}_f}{\text{BAG}_f} \end{cases}
\]

3 BANDWIDTH RESERVATION METHODS

In this section, we first describe the optimization problem associated to the bandwidth reservation problem of the BLS shapers. Afterwards, we detail the two proposed methods to solve such a problem: Heuristic Deadline and Dichotomous Deadline methods.

3.1 Problem formulation

The aim is to find reserved BLS bandwidth within each multiplexer \( \text{mux} \) along the path of each flow \( f \in \text{RC} \), minimizing the end-to-end delay bound of RC flow \( f \), while fulfilling the following constraints:

(i) the **class rate constraint**, stating that in each output port \( \text{mux} \), the input rate of an aggregate traffic class \( j \) must be lower than the minimum guaranteed service rate, denoted \( b_j^{\text{mux}} \);

(ii) the **aggregate rate constraint**, stating that the load of an output port multiplexer is lower than the output link capacity \( C \);

(ii) the **deadline constraint**, stating that the end-to-end delay bound of a flow \( f \) in class \( j \) (\( \text{EED}_j, f \)) must be lower than its deadline \( D_l \).

If we consider testing \( N \) values for each BLS parameter \( (L_{\text{M}}^{\text{mux}}, L_{\text{R}}^{\text{mux}}, \text{BW}^{\text{mux}}) \) within \( m \) output ports for \( l \) flows, then we have a complexity of \( O(m \cdot N^3 \cdot m) \) for the whole network. In order to drastically reduce such a complexity, we will minimize the delay bound for the aggregate traffic in RC class within each output port, instead of conducting one minimisation per-path per-flow; thus reducing the complexity down to \( O(m \cdot N^3) \).

Hence, we define a local deadline (resp. local delay bound) for class \( j \) in each output port multiplexer \( \text{mux} \), denoted \( D_{\text{local}}^{\text{mux}} \) (resp. \( \text{delay}_{\text{local}}^{\text{mux}} \)), that has to be fulfilled by the aggregate traffic class \( j \in \{\text{SCT, RC} \} \) in the output port \( \text{mux} \):

\[
D_{\text{local}}^{\text{mux}} \geq \text{delay}_{\text{local}}^{\text{mux}}(L_{\text{M}}^{\text{mux}}, L_{\text{R}}^{\text{mux}}, \text{BW}^{\text{mux}})
\]

Consider \( r_{\text{MUX}} \) the set of flows of a class \( j \) in an output port multiplexer \( \text{mux} \), the relaxed optimisation problem can be formulated as follows:

\[
\forall \text{mux}, \quad \min_{L_{\text{M}}^{\text{mux}}, L_{\text{R}}^{\text{mux}}, \text{BW}^{\text{mux}}} \text{delay}_{\text{RC}}^{\text{mux}}(L_{\text{M}}^{\text{mux}}, L_{\text{R}}^{\text{mux}}, \text{BW}^{\text{mux}})
\]

s.t. \( \forall \text{mux} \), \( \forall j \in \{\text{SCT, RC} \} : \)

(i) \( r_{f, \text{MUX}}^{\text{mux}} \geq \sum_{f \in j} r_f \)

(ii) \( \sum_{g \in \{\text{SCT} \}} r_g + \sum_{f \in \{\text{RC} \}} r_f \leq C \)

(iii) \( D_{\text{local}}^{\text{mux}} \geq \text{delay}_{\text{local}}^{\text{mux}}(L_{\text{M}}^{\text{mux}}, L_{\text{R}}^{\text{mux}}, \text{BW}^{\text{mux}}) \)

(2)

Based on the guaranteed service curves for RC and SCT classes within \( \text{mux} \) defined in [3] and the basic Network Calculus theorem to compute the delay bound in each \( \text{mux} \), i.e., the delay bound is the maximum horizontal distance between the arrival and service curves, the derived optimisation problem in (2) is a non-linear problem, with complex functions defining the delay bounds. There are many ways of solving such a problem numerically, such as brute force method, random search or heuristics. In our case, we will solve this problem based on heuristics taking advantages from conducted sensitivity analysis of our analytical model, which is not detailed in this paper due to the lack of space.

3.2 Solving the problem

**Computing \( L_R^{\text{mux}} \)**

The value of \( L_R^{\text{mux}} \) maximizing the minimum service rate of SCT defined in [3], while limiting the impact on RC traffic, is as follows:

\[
L_R^{\text{mux}} = \text{MFS}_{RC} \cdot \text{BW}^{\text{mux}}
\]

**Computing \( L_M^{\text{mux}} \)**

The delay bound of RC class within \( \text{mux} \) defined in [3] is as follows:

\[
delay_{\text{RC}}^{\text{mux}}(L_{\text{M}}^{\text{mux}}, L_{\text{R}}^{\text{mux}}, \text{BW}^{\text{mux}}) = \frac{A}{(1 - \text{BW}^{\text{mux}}) \cdot C} + \frac{\text{MFS}_{SCT}}{C} - \frac{\text{BW}^{\text{mux}}}{C} \cdot (\text{BW}^{\text{mux}} - \text{MFS}_{RC})
\]

(4)

where \( A = \text{BW}^{\text{mux}} + \text{MFS}_{RC} n_{RC}^{\text{mux}} + C - \text{MFS}_{RC} + \text{MFS}_{RC} \cdot \text{SCT} \cdot \text{RC} \times \{\text{SCT} \cap \text{RC} \} \times \text{MFS}_{RC} \) and \( n_{\text{links}} \) the number of input links sending RC flows to \( \text{mux} \).

To select the \( L_M^{\text{mux}} \) value minimising the delay function in Eq(4), we compute the null point of the derived delay function; thus:

\[
L_M^{\text{mux}} = \text{MFS}_{RC} \cdot \text{BW}^{\text{mux}} + \frac{A \cdot (1 - \text{BW}^{\text{mux}})}{\text{MFS}_{SCT} \cdot \text{BW}^{\text{mux}}}
\]

(5)

Therefore, giving Eq(3) and Eq(5), we have reduced the number of unknown BLS parameters within \( \text{mux} \) to only one, i.e., \( \text{BW}^{\text{mux}} \), to solve the optimisation problem in (2). To compute \( \text{BW}^{\text{mux}} \), we propose Algorithm 1, which takes into account as inputs \( D_{\text{local}}^{\text{mux}} \) and \( D^{\text{max}} \). We use a loop to compute the possible values for \( \text{BW}^{\text{mux}} \) in Line 2. Inside the loop, we compute the corresponding values of \( L_{\text{M}}^{\text{mux}} \), i.e., \( L_{\text{R}}^{\text{mux}} \), and \( L_{\text{M}}^{\text{mux}} \), i.e., \( L_{\text{M}}^{\text{mux}} \). Then, in Lines 3 and 4, we compute the SCT and RC delay bounds in Lines 5 and 6. Next, in Line 7, we verify the local deadlines conditions. If they are fulfilled, we store the delays and bw in Outputs, in Line 8. Finally, after testing all bw in the loop, we select the \( \text{BW}^{\text{mux}} \) leading to the minimum RC delay bounds, in Line 12. If no \( \text{BW}^{\text{mux}} \) fulfills the condition, we return \( +\infty \) for each delay bound. As we can notice, we need to define both local deadlines of SCT and RC within \( \text{mux} \) to enable Algorithm 1.

Hence, we have defined two methods to compute these local deadlines: Heuristic Deadline (HD) and Dichotomous Deadline (DD) methods.
Algorithm 1 BLS Bandwidth Reservation algorithm in a multiplexer $mux$ knowing the local deadlines: BLSparsams()

Require: $D_{SCT}^{mux}$, $D_{RC}^{mux}$, $B_{SCT}^{mux}$, $MFS_{SCT}$; $l_{mux}$, $l_{RC}$, $f_{mux}$, $f_{RC}$, $MFS_{RC}$.

Ensure: $BW_{mux}$ delay$_{SCT}$ delay$_{RC}$.

1: Data: $[D_{SCT}^{mux}$, $D_{RC}^{mux}$, $B_{SCT}^{mux}$, $MFS_{SCT}$; $l_{mux}$, $l_{RC}$, $f_{mux}$, $f_{RC}$, $MFS_{RC}$].
2: for $bw$ in $[0.001 : 0.001 : 0.999]$ do
3: $l_{mux} = MFS_{RC} · bw$
4: $l_{RC} = l_{mux}$ in Eq.5
5: $D_{mux} = Delay_{max}^{mux}(Data, bw, l_{mux}, l_{RC})$
6: $D_{RC} = Delay_{max}^{RC}(Data, bw, l_{mux}, l_{RC})$
7: if $D_{mux} ≤ D_{RC}^{mux}$ and $D_{mux} ≤ D_{RC}^{mux}$ then
8: Outputs.add($bw, d_{RC}^{mux}, d_{SCT}^{mux}$)
9: end if
10: end for
11: if notEmpty(Outputs) then
12: Outputs.get(IndexOfMinDRCs(Outputs))
13: else
14: $[-, +∞, +∞]$ %no admissible parameters
15: end if

Dichotomous Deadline Method

To compute the delay bound of class $SCT$ for each output port multiplexer $mux_i$ along the path of a flow $f$, we will use two deadline values: one leading to delay bounds equal or lower to the deadline, and one leading to delay bounds equal or higher than the deadline.

The first deadline is computed with the Heuristic Deadline method, and may lead to a lower $SCT$ delay bound:

$$D_{under,mux_i}^{SCT} = \frac{\sum_{f_{mux_i}} r_{f_{pw}} \cdot \sum_{mux_{path}} D_{SCT}^{mux_i}}{\sum_{mux_{path}} D_{SCT}^{mux_i}}$$

To obtain a higher $SCT$ delay bound, we consider the following $SCT$ deadline:

$$D_{over,mux_i}^{SCT} = \sum_{mux_{path}} D_{SCT}^{mux_i}$$

The Dichotomous Deadline (DD) method is detailed in Algorithm 2.

From Line 2 to Line 10, we initialise the dichotomous search. In Lines 3, 4 and 5, we compute the initial Deadlines, i.e., $D_{SCT}^{mux}$, $D_{RC}^{mux}$ and $D_{under,mux}^{SCT}$ for each $mux$. This leads in Lines 6 and 7 to the computation of the corresponding $SCT$ delay bounds $delay_{over,mux}^{SCT}$ and $delay_{under,mux}^{SCT}$ using Algorithm 1. After all the $mux$ have been considered, we compute in Lines 9 and 10 the two dichotomous variables: $\sum D_{SCT}^{mux}$ and $\sum D_{SCT}^{over}$. Then in Line 11, we check whether the value $\sum D_{mux_{path}}^{over,mux} delay_{over,mux}^{SCT}$ is actually lower than $\sum D_{mux_{path}}^{SCT}$. If not, we return $D_{over,mux}^{SCT}$ since a dichotomous search is not possible. Else, we start the dichotomous search, where $\sum D_{SCT}^{max}$ is bounded by $\sum D_{mux_{path}}^{under,mux} delay_{under,mux}^{SCT}$ and $\sum D_{mux_{path}}^{under,mux} delay_{under,mux}^{SCT}$.

In Line 14, we set the stop condition using an $\epsilon < 0$ such as: $\sum D_{mux}^{SCT} - \sum D_{mux_{path}}^{under,mux} delay_{SCT}^{SCT}. Then, we start the next iteration in Line 15, by computing the current $\sum D_{mux_{path}}^{over,mux}$. We use it to compute the local deadlines in each $mux$ and the resulting SCT delay bounds in Lines 17 and 18.

The final steps consist in determining whether $\sum D_{mux_{path}}^{over,mux} delay_{SCT}^{SCT}$ (the sum of the current delay bounds) is lower or higher than $\sum D_{mux}^{SCT}$ in Line 20. Then, we redefine the values of the current loop, either $\sum D_{mux_{path}}^{over,mux}$ in Line 21, or $\sum D_{mux_{path}}^{under,mux}$ and $D_{mux_{path}}^{under,mux}$ in Lines 23 and 24.

4 PERFORMANCE EVALUATION

In this section, we first describe our case study. Afterwards, we assess the efficiency of the introduced bandwidth reservation methods to enhance the extended AFDX performance, in comparison to an intuitive method, since there is no other existing methods.

4.1 Case study

Our case study is a representative avionics communication architecture of the A380, based on a 1-Gigabit AFDX backbone network.
In this multi-hop network, each end-system \( e \) generates \( n^e_i \) flows of type \( i \in \{SCT, RC, BE\} \). We consider that all end-systems are identical and each generates the same number of flows \( n^e_i \).

As a consequence, the utilisation rate in both the first and second switches is the bottleneck utilisation rate (i.e. maximum utilisation rate along a path) for each type of traffic \( i \in \{SCT, RC, BE\} \),

\[
U^i_{bn} = 16 \cdot \frac{MF_{SCT}}{BW}.
\]

We consider the traffics SCT, RC and BE defined in Table 2 and the scenarios detailed in Table 3.

Concerning the intuitive method, we consider the following parameters: we set the reserved bandwidth to the bottleneck utilisation rate \( BW = U^i_{bn} \cdot SCT \), and \( L_R = MF_{SCT} \cdot BW \) and \( L_M = 80 \cdot MF_{SCT} \cdot (1 - BW) \), to enable the transmission of a maximum SCT burst of 80 frames within the BLS, i.e., a generated burst of 5 SCT flows per end-system.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Traffic type</th>
<th>MF (Bytes)</th>
<th>BW (ms)</th>
<th>deadline (ms)</th>
<th>jitter (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/2</td>
<td>SCT</td>
<td>64</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>RC</td>
<td>320</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>BE</td>
<td>1024</td>
<td>8</td>
<td>none</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 2: Avionics flow Characteristics

### 4.2 Numerical Results

In this section, we study the impact of optimized BLS bandwidth compared to the intuitive one on a multi-hop network. Hence, we compare the delay bounds of SCT and RC based on the extension of the timing analysis in [3] to the multi-hop case, under HD and DD methods in reference to the intuitive method. The results for the two scenarios are illustrated in Figures 4 and 5. It is worth noting that we only present the admissible results, i.e., when all the deadlines are fulfilled.

First, considering the maximum bottleneck utilisation rate:

(i) in Figure 4, we note that the maximum bottleneck SCT utilisation rate is 32% with the intuitive method, 36% with HD method, and 40% with DD method;

(ii) in Figure 5, the maximum bottleneck RC utilisation rate is 28% with the intuitive method, and 40% with both HD and DD methods.

These results show an improvement of the SCT (resp. RC) scheduling up to 25% (resp. 42%) under the optimized bandwidth reservation methods, in comparison to the intuitive one.
Secondly, in Figure 4(b), the RC delay bounds with HD method are lower than the delay bounds with the intuitive one until $U_{RC} = 18\%$. However, for $U_{RC}$ between 18\% and 32\%, the intuitive method is better than HD one. The same issue is visible in Figure 5(b) for $U_{RC}$ between 20\% and 28\%.

To understand the reasons of this issue, in Figure 6, we present separately the SCT delay bound in the first and in the second switch output ports, denoted $delay^{mux1}_{SCT}$ and $delay^{mux2}_{SCT}$. In Figure 6(a), we can separate the SCT delay bounds under HD method in 4 areas, with $U_{SCT}$:

(i) between 0\% and 20\%, $delay^{mux1}_{SCT} < D^{mux1}_{SCT}$ and $delay^{mux2}_{SCT} < D^{mux2}_{SCT} \Rightarrow EED_{SCT} < D_{SCT}$. The multiplexer deadline is reached in neither switches;

(ii) between 20\% and 45\%, $delay^{mux1}_{SCT} = D^{mux1}_{SCT}$, $delay^{mux2}_{SCT} < D^{mux2}_{SCT} \Rightarrow EED_{SCT} < D_{SCT}$. The switch output port 1 deadline (about 1ms, see Figure 6(a)) is reached and the SCT delay bound remains at this deadline $D^{mux1}_{SCT}$ until $U_{SCT} = 50\%$ (see Figure 6(a)). However, in the second switch output port, the SCT delay remains firmly below its deadline;

(iii) between 45\% and 50\%, $delay^{mux1}_{SCT} = D^{mux1}_{SCT}$, $delay^{mux2}_{SCT} = D^{mux2}_{SCT} \Rightarrow EED_{SCT} = D_{SCT}$. The SCT end-to-end delay bound is equal to the end-to-end deadline, as the delays in both output ports are equal to their respective deadlines (see Figure 6);

(iv) between 50\% and 80\%, $delay^{mux1}_{SCT} > D^{mux1}_{SCT}$ and $delay^{mux2}_{SCT} = D^{mux2}_{SCT} \Rightarrow EED_{SCT} > D_{SCT}$. The end-to-end delay bound is higher than the end-to-end deadline as the delay in the switch output port 1 is higher than its deadline (see Figure 6).

Hence, this highlights the fact that limiting the local deadline in an output port without taking into account the state of other ones in the path decreases the performance of the RC delay bounds.

Contrary to the HD method, the DD method takes into account the output ports along the flow path. In Figures 4 and 5, the RC delay bounds with DD method are better than the ones with both intuitive and HD methods. For instance at $U_{SCT} = 32\%$, the RC delay bound is improved by 49\% with reference to the intuitive one, and by 74\% compared to HD method.

We can conclude from these results that Dichotomous Deadline method leads to a great improvement over both the intuitive and Heuristic Deadline methods. The schedulability of SCT is actually increased by up to 31\% and the RC delay bound is decreased by up to 75\%. Nevertheless, the Dichotomous Deadline method needs much higher computation times, in comparison to Heuristic Deadline method, e.g., up to 10 times.

5 CONCLUSIONS

In this paper, we have proposed two optimized bandwidth reservation methods for TSN/BLS shapers in an extended AFDX, denoted Heuristic Deadline and Dichotomous Deadline methods, to enhance the network performance, in terms of schedulability and delay bounds. The conducted performance evaluation on a realistic avionics case study highlights the benefit of using such methods, and particularly the Dichotomous Deadline method. The latter leads to a great improvement over both the intuitive and Heuristic Deadline methods, in terms of schedulability (up to 31\% for SCT) and delay bounds (up to 75\% for RC), but at the expense of higher computation times, e.g., up to 10 times.

As a next step, we will generalize such analyses to an extended AFDX with multiple BLS classes to offer higher configuration flexibility.

REFERENCES


