

Robust Priority Assignment applied to Avionics Full-Duplex Switched Ethernet Networks

Edison Pignaton de Freitas^{1 2} and Marcelo Antonio Azeredo da Rocha²

¹*School of Information Technology, Halmstad University, Halmstad, Sweden*

E-mail: edison.pignaton@hh.se

²*Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*

E-mail: marcelo.azedorochoa@inf.ufrgs.br

Abstract

The priority assignment of flows on an Avionics Full-Duplex Switched Ethernet (AFDX) network impacts directly the end-to-end delay of all messages. This paper proposes the use of Robust Priority Assignment (RPA) in order to increase the additional interference that the most vulnerable virtual links of a AFDX network can support by decreasing their worst-case end-to-end delays through a better priority assignment. A comparison is performed with other state-of-the-art techniques used to assign priorities. The results shows that the use of RPA increases the network robustness in 307 in a sample configuration.

Keywords: AFDX, Robust Priority Assignment, Optimal Priority Assignment, Priority Assignment, Delay, Worst-Case End-to-End Delay, Network Robustness.

1 Introduction

The increasing complexity of avionics embedded systems has led to an increase in the number of devices installed and in the amount of data exchanged, thus increasing the number of wired connections between these systems. Known solutions, as ARINC-429 buses, add a significant amount of weight and are not fit for modern aircraft configurations. Avionics Full-Duplex Switched Ethernet (AFDX) networks represent a solution by multiplexing communication flows over a full-duplex switched Ethernet network, reducing the overall weight and increasing the bandwidth. However, the use of AFDX networks as the backbone for communication between end systems introduces a problem related to indeterminism at the switch level, given that different flows may arrive at the same time in the same node.

For certification reasons, it is mandatory to provide an upper bound for the end-to-end delay of a given frame carrying flight-critical data between end systems [1]. Several approaches were developed and improved over the years in order to calculate this worst-case delay. The first approach developed to estimate the end-to-end upper bound delay in the context of an AFDX network was the Network Calculus (NC) approach. As concluded by Yao and Zhu [2], this ap-

proach adds pessimism to the analysis, resulting in a calculated worst-case delay that is much larger than the one measured, leading to an underutilization of the network. Following the Network Calculus approach, the Trajectory Approach (TA) was applied in the analysis of the upper bound delay by Bauer et. al [3]. The authors used an optimized Trajectory Approach, which takes into account the serialization of frames, and concluded that, on average, the upper bound delay achieved through TA is less pessimistic than NC.

Currently, AFDX networks mostly use a First In First Out (FIFO) scheduling policy. However, priority assignment is essential to avoid hardware underutilization, to develop scalable systems and to avoid intermittent failures due to deadline misses [4]. Plus, the addition of other types of flows on AFDX networks, like audio and video, is envisioned in the future [5]. These different flows have different timing constraints and criticality levels. Therefore, the assignment of priorities for each flow may allow for better utilization of the network. Hamza et. al [6] proposed the priority assignment using the Optimal Priority Assignment (OPA). OPA is widely used in the context of real-time systems. This algorithm is based on a schedulability test that must respect a set of conditions. The authors proved that these conditions are respected when using OPA on AFDX networks, conclud-

ing that OPA can significantly reduce the worst-case end-to-end delay. However, OPA does not specify which tasks (or in this context, frames) should be tried at each priority level when calculating the optimal assignment. Knowing that the order of frames provided to the algorithm have influence over the priority assigned, there is a chance that a poor initial ordering may result in a priority assignment which leaves the system marginally schedulable. This problem was already addressed by Davis and Burns [7] through the development of the Robust Priority Assignment (RPA) algorithm, which is both optimal and robust (i.e., resulting in a priority assignment that is the most optimal and resilient regarding timing interference). This paper describes the applicability of RPA in order to assign robust priorities for AFDX's flows, comparing the worst-case upper bound delays (estimated using Trajectory Approach) achieved using RPA with the delay achieved through OPA. This paper's main contribution is to prove that RPA can be used in the context of AFDX networks, delivering lower worst-case delays than OPA and FP/FIFO solutions.

2 Related Works

Regarding delay estimation, Yao and Zhu [2] compared the delay bound calculated through Network Calculus and the one achieved simulating an AFDX network through a Simulink toolbox (TrueTime), which provides real-time simulation tools. The authors studied mixed-criticality avionics traffic for AFDX network, which can transmit both critical traffic and non-critical traffic. In their proposed architecture, critical traffic was scheduled by a Bandwidth Allocation Gap-based scheduler, and non-critical traffic was scheduled in a Round Robin manner. From the results, the authors concluded that the measured delay is much smaller than the estimation, confirming the pessimistic behavior of the Network Calculus.

Bauer *et al.* [3] proposed an improvement to the end-to-end delay analysis of an AFDX network by using the Trajectory Approach, comparing the results to those obtained by using Network Calculus for end-to-end delay analysis in a real AFDX configuration. The first contribution of this paper is an explanation of how the grouping technique can be introduced using the Trajectory approach. Moreover, a proof of the correctness of the corresponding computation was given. A second contribution was the comparison of worst-case delay upper bounds obtained by Network Calculus and the Trajectory Approach. The bounds obtained by Trajectory Approach were tighter and the authors concluded that this approach is less pessimistic than the Network Calculus one.

Li and George [8] studied how to compute worst-case end-to-end delays of flows sent on an AFDX FIFO network, focusing on the Trajectory approach, known to provide tight worst-case end-to-end delay estimations. However, the authors stated that this approach can lead to optimistic end-to-end delays, thus leading to certification issues. Therefore, their goal was to characterize this optimism problem and to provide a solution to it. The authors illustrated that this optimism problem is induced by a problem on the computation of the serialization factors for flows sent on the same link. The error rate found on the worst-case delay upper bound in our examples

can reach 10%. Finally, they proposed a solution to correct the optimism problem in the general case.

Regarding priority assignment, Leung and Whitehead [9] proved that the problem of priority ordering and feasibility is NP-hard. For synchronous systems, the authors proved that the complexity of priority assignment is $O(n \log 2n)$, equivalent to the amount of tasks. Therefore, the NP-hardness comes from the feasibility testing. For asynchronous systems, the authors were not able to determine the complexity of finding a priority order and stated that one must test for all $n!$ combinations.

Audsley [10] addressed the problem of static priority assignment for real-time asynchronous systems. Through the development of an algorithm, the author has shown that $n(n+1)/2$ priority orderings must be tested in order to either find a feasible priority ordering or to determine the system as unschedulable.

Davis and Burns [4] proved that three conditions are sufficient and necessary for Audsley's algorithm to provide optimal priority assignment. In another work [7], the authors addressed the problem related to the fact that Audsley's algorithm is heavily dependent on the initial ordering of tasks, which may influence in the priority ordering returned from Audsley's algorithm, creating a priority ordering marginally schedulable. In order to overcome this problem, the authors developed the RPA algorithm, which returns a priority ordering that is able to tolerate a given amount of additional interference.

Hamza *et al.* [6] studied the assignment of priority to flows on an AFDX network using Audsley's OPA algorithm. As schedulability test, the authors used the Trajectory Approach. The authors proved that the assignment of priorities to AFDX flows is OPA-compatible by proving that all three conditions, stated by Davis and Burns [4], are respected. Through the use of OPA, the authors presented a decrease in the worst-case delay up to 30%. However, the author did not limit the number of priority levels that could be used, and their results were based on more priority levels than are available in AFDX switches currently.

Regarding performance analysis, Suthaputchakun *et al.* [11] studied the performance analysis of AFDX based on aspects such as the impact of high network load, loose time synchronization, and unexpected events. The authors concluded that higher traffic does not impact the overall system performance. However, the authors found that unexpected events can impact the average end-to-end jitter and reception rate.

3 Context

3.1 AFDX Network

The Avionics Full Duplex Switched Ethernet is a specific implementation of ARINC 664 part 7. In an AFDX network, End Systems are interconnected and exchange frames through Virtual Links (VL), which are statically defined unidirectional communication channels, defining a logical unidirectional connection between End Systems [12]. Fig. 1 depicts an example of an AFDX network configuration.

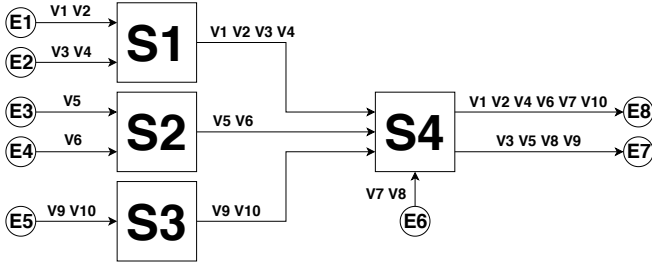


Figure 1: Example of an AFDX Network.

Virtual Links are defined by their Bandwidth Allocation Gap (BAG), which is the minimum duration between two consecutive frames. Another important aspect of a virtual link is the minimum and maximum frame length, s_{min} and s_{max} respectively. The switches are defined by a FIFO buffer for each output port. Each AFDX switch has its switching delay upper bounded. For each virtual link a fixed priority (high or low in the AFDX context) is assigned. In this way, a given frame may be delayed by the transmission of frames with higher priority and also by the competition with frames of the same priority, in addition to the switching delay added at each switch crossed.

3.2 Trajectory Approach with serialization

The Trajectory Approach [13] represents a way to determine the worst-case end-to-end delay in distributed systems. The Trajectory Approach considers a given set of sporadic flows in given network topology of interconnected nodes [3]. According to Bauer *et al.* [14], the Trajectory Approach considers a given packet m , generated at time t and from flow i . In this way, the Trajectory Approach will identify all the competing packets which will impact in the end-to-end delay of packet m (by checking all nodes visited by m). This calculation is performed from the last visited node to the first visited node in order to compute the last starting time of m on its last node [14]. In summary, the Trajectory Approach will compute the delay that all frames from virtual links which are crossing paths with the virtual link under analysis may create in order to compute the worst-case end-to-end delay. The Trajectory Approach is based on the busy period concept and takes into account the set of virtual links with the same priority, with higher priority and also with lower priority (given the nonpreemption concept of AFDX, lower priority virtual links may delay higher priority messages) in order to compute the final end-to-end delay. The serialization concept, integrated successfully in the Trajectory Approach by Bauer *et al.* [3], states that frames from virtual links with the same source cannot arrive at a given destination at the same time. Therefore, the Trajectory Approach simplifies the network topology in order to assume that only the frame that can cause the biggest delay at a given node will be taken into account, removing all other frames from the same source but with smaller impact on the worst-case end-to-end delay of the virtual link under analysis. The complete mathematical formulation of the Trajectory Approach applied to AFDX networks was defined by Bauer *et al.* [14].

3.3 Optimal Priority Assignment

The main goal of a policy to assign priority to tasks (or frames) is to provide a schedulable order whenever such order exists [4]. In this way, Audsley [10] addressed the problem of priority assignment for asynchronous task sets by presenting an algorithm called Optimal Priority Assignment (OPA). Using a polynomial number of schedulability tests, OPA produces an optimal priority ordering if such an order exists. The algorithm performs, at most, $n(n+1)/2$ schedulability tests in order to return a viable priority assignment (or to return that there is no schedulable order). A specific schedulability test is used in order to assess if, based on the current priority assigned to a given task, its deadline will be met. In order to choose a schedulability test to be used in the context of OPA, three conditions must be respected [4]:

- 1) For a given schedulability test, the schedulability of a specific task may be dependent on a set of tasks with higher priority. However, it may never be dependent on their relative priority ordering.
- 2) For a given scheduling test, the scheduling of a specific task may depend on a set of tasks with lower priority. However, it may never be dependent on their relative priority ordering.
- 3) When the priority of any two tasks of adjacent priorities is changed, the task previously schedulable at a lower priority cannot become unschedulable when assigned a higher priority.

The applicability of the trajectory approach as the schedulability test used within the OPA algorithm is detailed by Hamza *et al.* [6]. In summary:

- 1) The worst-case end-to-end delay of a frame in a given virtual link is dependent only on the set of higher priority virtual links, but not on their relative priority ordering.
- 2) The worst-case end-to-end delay of a frame in a given virtual link may be dependent on a set of lower priority virtual links, but not on their relative priority ordering.
- 3) When the priority of any two virtual links of adjacent priorities is swapped, the virtual link previously schedulable at a lower priority will not become unschedulable when assigned a higher priority (i.e. the worst-case end-to-end delay of the virtual link being assigned a higher priority will not increase).

OPA assigns one priority level per task in order to find a schedulable priority ordering. However, in several applications, there are only a few priority levels available (e.g. AFDX, where there are only two priority levels available). This problem was addressed [10] by modifying the original OPA algorithm in order to minimize the number of priority levels used to find a feasible ordering. This minimization is achieved through the maximization of tasks assigned per priority level, from the lowest to the highest priority available. The algorithm for the Optimal Priority Assignment (minimizing priority levels) is described below.

```

for each priority level  $i$ , lowest first do
  for each unassigned task  $\tau$  do
    if  $\tau$  is schedulable at priority  $i$  assuming all
      unassigned tasks have higher priorities then
      | assign priority  $i$  to task  $\tau$ 
    end
  end
  if no tasks are schedulable at priority  $i$  then
  | return unschedulable
  end
  if no unassigned task remain then
  | break
  end
end
return schedulable

```

Algorithm 1: Optimal Priority Assignment with Minimal Priority Levels

A significant drawback of OPA is the fact that the algorithm does not specify the order in which tasks should be tried at each priority level. Therefore, given an unfavorable initial ordering, there is a chance that the priority ordering resulting from OPA will be only marginally stable [7].

3.4 Robust Priority Assignment

Audsley's OPA algorithm makes an arbitrary choice regarding which schedulable tasks must be assigned at each priority level [7]. Therefore, it's possible that the priority assignment may be marginally stable, fragile to any minor changes or underestimations. Davis and Burns [4] addressed this problem by developing the RPA. The RPA assumes the presence of an additional interference modelled as a function $E(\alpha, \omega, i)$, where α is a scaling factor used to model the variability in the amount of interference, ω is the time interval where the interference occurs and i is the priority level affected by the interference. As an example [7], a system subjected to an additional interference from an interrupt handler of indeterminate duration activated, at most, every 100 μs , affecting all priority levels, will have its interference function as follows:

$$E(\alpha, \omega, i) = \alpha \left(\frac{\omega}{100} \right) \quad (1)$$

where ω represents the time interval, in μs , where the interrupt handler is activated. Given that the function affects all priority levels, there is no influence of i over the interference function. In summary, of all schedulable priority orderings, RPA outputs the ordering where the highest additional interference is tolerated [7]. Given that RPA is based on the OPA algorithm, it requires the same three conditions for the schedulability test. In this way, any schedulability test OPA-compatible is automatically RPA compatible, as is the case with the Trajectory Approach [6]. The goal of RPA is to maximize the possible additional interference for the most vulnerable task (i.e. the task closest to not meet its deadline), given that a system is only as robust to additional interference as its most fragile task. The RPA algorithm is detailed below.

```

for each priority level  $i$ , lowest first do
  for each unassigned task  $\tau$  do
    determine the largest value of additional
      interference for which task  $\tau$  is schedulable at
      priority  $i$  assuming that all unassigned tasks
      have higher priority
    end
  if no tasks are schedulable at priority  $i$  then
  | return unschedulable
  else
  | assign all tasks that tolerates the maximum
    additional interference at priority  $i$  to priority  $i$ 
  end
end
return schedulable

```

Algorithm 2: Robust Priority Assignment

In the scope of the present work, the RPA algorithm was modified in order to return a schedulable priority ordering only if the number of priority levels used was smaller than or equal to two priority levels, in order to comply with the number of priorities of AFDX switches. Also, by assigning all tasks that tolerates the maximum additional interference at priority i to priority i (instead of assigning the task that tolerates the most additional interference only), there is a decrease in the number of priority levels used. This modification was based on Audsley's algorithm for decreasing the number of priority levels used for optimal priority assignment.

4 Comparative Analysis: Results and Discussion

The comparative analysis between the priority assignment using both OPA and RPA is conducted in three steps: first, the priority ordering of all virtual links in the proposed network topology is assigned by using OPA, and the worst-case end-to-end delay of all virtual links is calculated using the Trajectory Approach. Then, the same approach is applied, but using RPA instead of OPA. Lastly, the worst-case delays and the amount of additional interference supported (i.e. worst-case end-to-end delay subtracted from the deadline) from both priority ordering strategies are compared. In this way, it's possible to compare whether the RPA is indeed increasing the system's robustness. The network topology used in the comparative analysis is detailed in Fig. 1. The details regarding each virtual link are described in Table I.

The priorities assigned to each virtual link and the worst-case end-to-end delay (WCETED) of each virtual link are presented in Table 2.

The assignment of priorities resulted from RPA decreasing the end-to-end delay. There was no increase in the end-to-end delay of any virtual link. A comparison between the maximum interference supported when assigning priorities using both OPA and RPA is presented in Fig. 2.

The system as a whole became more robust by using RPA to assign priorities instead of OPA, given that V1, V2, V3, V4, V5, and V6 can now handle an interference of 53 μs without missing its deadline. Through the OPA, these virtual links

Table 1
Parameters used in the comparative analysis.

Virtual Link	BAG (μ s)	Smax (bits)	Deadline (μ s)
V1	4000	4000	325
V2	4000	4000	325
V3	4000	4000	325
V4	4000	4000	325
V5	4000	4000	325
V6	4000	4000	325
V7	4000	4000	325
V8	4000	4000	325
V9	4000	4000	325
V10	4000	4000	325

Table 2
Priority assignment and WCETED of OPA and RPA.

Virtual Link	Priority (OPA)	Priority (RPA)	WCETED (OPA) (μ s)	WCETED (RPA) (μ s)
V1	LOW	HIGH	312	272
V2	LOW	HIGH	312	272
V3	LOW	HIGH	312	272
V4	LOW	HIGH	312	272
V5	LOW	HIGH	312	272
V6	LOW	HIGH	312	272
V7	LOW	LOW	216	216
V8	LOW	LOW	216	216
V9	LOW	LOW	272	272
V10	LOW	LOW	272	272

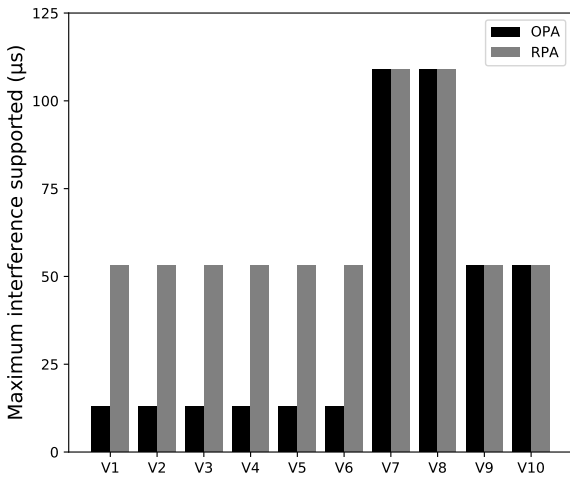


Figure 2: Maximum interference supported when ordering using OPA and RPA (+40 μ s robustness gain in V1-V6 for RPA).

could handle only 13 μ s of interference. However, RPA does not guarantee that the system will become more robust (i.e. not all of the most vulnerable virtual links will always have their end-to-end delay decreased), as it only guarantees that the most robust priority ordering will be found. As an example of this behavior, a new comparative analysis will be performed. The deadlines of V1 and V6 are decreased in or-

der to change the priority ordering achieved through OPA and RPA. However, the network topology and all other parameters used in the first comparative analysis are still the same. The parameters used in this new comparative analysis are summarized in Table 3.

Table 3
New parameters used in the comparative analysis (decreasing deadlines of V1 and V6).

Virtual Link	BAG (μ s)	Smax (bits)	Deadline (μ s)
V1	4000	4000	300
V2	4000	4000	325
V3	4000	4000	325
V4	4000	4000	325
V5	4000	4000	325
V6	4000	4000	300
V7	4000	4000	325
V8	4000	4000	325
V9	4000	4000	325
V10	4000	4000	325

After running both OPA and RPA algorithms again using the Trajectory Approach as schedulability test for the new parameters presented in Table 3, the priority orderings and worst-case end-to-end Delay are summarized in Table 4.

Table 4
Priority assignment and WCETED of OPA and RPA after decreasing deadlines of V1 and V6.

Virtual Link	Priority (OPA)	Priority (RPA)	WCETED (OPA) (μ s)	WCETED (RPA) (μ s)
V1	HIGH	HIGH	272	272
V2	LOW	LOW	312	312
V3	LOW	HIGH	312	272
V4	LOW	LOW	312	312
V5	LOW	HIGH	312	272
V6	HIGH	HIGH	272	272
V7	LOW	LOW	216	216
V8	LOW	LOW	216	216
V9	LOW	LOW	272	272
V10	LOW	LOW	272	272

The assignment of priorities resulted from RPA decreased the end-to-end delay of V3 and V5 only. There was no increase in the end-to-end delay of any virtual link. A comparison between the maximum interference supported when assigning priorities using both OPA and RPA, using the new parameters with decreased deadlines for both V1 and V6, is presented in Fig. 3. While the maximum interference supported by V3 and V5 increased, the overall interference supported by the system remained the same (13 μ s).

In this case, the most robust priority assignment was achieved, but the overall robustness of the system could not be increased. While RPA remains a viable strategy, the parameters of the network themselves must be analyzed (e.g., re-routing virtual links) in order to enhance robustness. To address this limitation, we propose a deterministic Robustness Booster

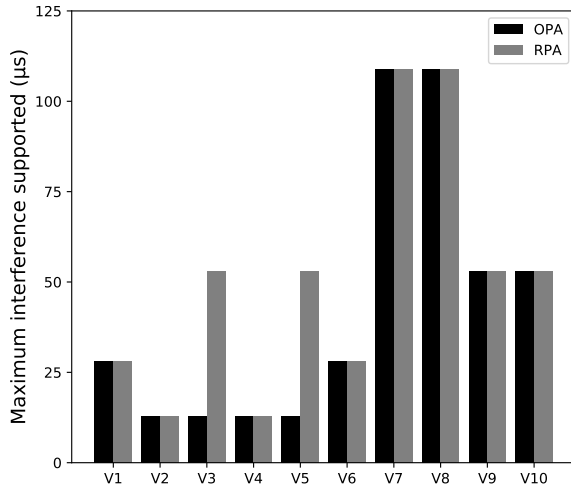


Figure 3: Maximum interference supported when ordering using OPA and RPA with the new parameters (+40μs robustness gain in V3 and V5 for RPA).

Module (RBM) that, after RPA, redistributes slack through local priority promotions, project-allowed parametric adjustments (e.g., BAG/Deadlines), and neighborhood pair swaps, accepting only changes that increase the minimum margin of the target flows without violating schedulability. This module is auditable and compatible with certification. Nevertheless, the notion of “absolute robustness” is constrained by physical and combinatorial limitations of the problem; therefore, we adopt the maximization of a robustness radius (max–min) as a practical objective.

5 Conclusion

The assignment of priorities through RPA has increased system robustness, as the worst-case end-to-end delay of the most vulnerable virtual links was reduced. Compared to OPA, RPA allowed 40 μs (307%) more interference to be supported under the same conditions, making it a potentially better fit for AFDX networks.

However, RPA does not always maximize interference support, since network parameters and topology also affect robustness. Moreover, the most robust assignment is not always ideal. Some virtual links carry critical data, and their robustness can be more important than that of the overall system. Future work may focus on improving robustness for selected virtual links to ensure their delays are minimized.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, partly by the Brazilian National Council for Scientific and Technological Development (CNPq) with the Project 311773/2023-0, and by the ELLIIT Strategic Research Network, Sweden.

References

- [1] Nassima Benammar, Frédéric Ridouard, Henri Bauer, and Pascal Richard. Forward end-to-end delay for afdx networks. *IEEE Transactions on Industrial Informatics*, 14(3):858–865, 2017.
- [2] Jianguo Yao, Jiahong Wu, Qingchun Liu, Zhiyong Xiong, and Guchuan Zhu. System-level scheduling of mixed-criticality traffics in avionics networks. *IEEE Access*, 4:5880–5888, 2016.
- [3] Henri Bauer, Jean-Luc Scharbarg, and Christian Fraboul. Improving the worst-case delay analysis of an afdx network using an optimized trajectory approach. *IEEE Transactions on Industrial Informatics*, 6(4):521–533, 2010.
- [4] Robert I Davis and Alan Burns. Improved priority assignment for global fixed priority pre-emptive scheduling in multiprocessor real-time systems. *Real-Time Systems*, 47:1–40, 2011.
- [5] Jean-Luc Scharbarg, Frédéric Ridouard, and Christian Fraboul. A probabilistic analysis of end-to-end delays on an afdx avionic network. *IEEE transactions on industrial informatics*, 5(1):38–49, 2009.
- [6] Tasnim Hamza, Jean-Luc Scharbarg, and Christian Fraboul. Priority assignment on an avionics switched ethernet network (qos afdx). In *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*, pages 1–8. IEEE, 2014.
- [7] Robert I Davis and Alan Burns. Robust priority assignment for fixed priority real-time systems. In *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*, pages 3–14. IEEE, 2007.
- [8] Xiaoting Li, Olivier Cros, and Laurent George. The trajectory approach for afdx fifo networks revisited and corrected. In *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 1–10. IEEE, 2014.
- [9] Joseph Y-T Leung and Jennifer Whitehead. On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance evaluation*, 2(4):237–250, 1982.
- [10] Neil C Audsley. On priority assignment in fixed priority scheduling. *Information Processing Letters*, 79(1):39–44, 2001.
- [11] Chakkaphong Suthaputthakun, Zhili Sun, Christoforos Kavadias, and Philippe Ricco. Performance analysis of afdx switch for space onboard data networks. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1714–1727, 2016.
- [12] Meng Li, Michaël Lauer, Guchuan Zhu, and Yvon Savaria. Determinism enhancement of afdx networks via frame insertion and sub-virtual link aggregation. *IEEE*

Transactions on Industrial Informatics, 10(3):1684–1695, 2014.

- [13] Steven Martin and Pascale Minet. Schedulability analysis of flows scheduled with fifo: application to the expedited forwarding class. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, pages 8–pp. IEEE, 2006.
- [14] Henri Bauer, Jean-Luc Scharbarg, and Christian Fraboul. Applying trajectory approach with static priority queuing for improving the use of available afdx resources. *Real-time systems*, 48:101–133, 2012.