A Tight End-to-End Delay Bound and Scheduling Optimization of an Avionics AFDX Network

Melhem Tawk¹, Guchuan Zhu¹, Yvon Savaria¹, Xue Liu², Jian Li³, and Fei Hu³

¹ ´École Polytechnique de Montréal
P.O. Box 6079, Station Centre-Ville, Montreal, QC, Canada H3C 3A7
² McGill University
3480 University Street, Montreal, QC, Canada H3A 2A7
³ Shanghai Jiao Tong University
800 Dongchuan Road, Minhang, Shanghai, China 200240

Abstract—Inspired by the response time analysis (RTA) technique, this paper presents a new method for the estimation of end-to-end delay bounds in AFDX networks. For the purpose of comparison, an end-to-end delay analysis based on the Network Calculus (NC) is first carried out. In order to apply the RTA technique, an interference model of virtual links with common transmission paths in a typical AFDX network configuration has been established. The corresponding worst-case response time (WCRT) as well as a schedulability criteria are then derived. Furthermore, a procedure for scheduling policy optimization is deduced based on RTA. Numerical analysis shows that the RTA can achieve a tighter end-to-end delay bound compared to the one given by NC. Simulations are also carried out to confirm the validity of the proposed approach.

Index Terms—AFDX, Delay bound, Response-time analysis, Network calculus, Real-time scheduling.

I. INTRODUCTION

The AFDX standard imposes a very strict determinism assessment requiring that any such system must provide services with a firm, mathematically provable, upper bound guarantee on end-to-end frame transit delay (see Section 3.1 in [1]). Therefore computation and analysis of transmission delays represent a challenging task in the design and the development of AFDX networks. In the literature, a considerable effort has been dedicated to this issue and different methods have been applied [2], [3], [4]. However, there is often an important gap between the estimated delay bound and the one obtained from experimentations. Therefore, finding tighter end-to-end delay bounds represents one of the central topics in the field of avionics communication systems.

One of the most popular tools employed in computing upper bounds of transmission delay in communication networks is the theory of Network Calculus (NC) [5], [6]. Essentially, this approach amounts to estimating the delay bound based on the characteristics of data flows and multiplexing mechanisms, described by, respectively, arrival curves and service curves. To determine the performance provided by a system, many work attempts to find tighter arrival curves with respect to fixed scheduling policies while assuming the schedulability [7], [8], [4], [9].

This paper presents an approach for upper bound estimation of end-to-end data transmission delay in multi-hop AFDX networks, which is inspired from a very popular tool in real-time systems, namely the worst case response time analysis [10], [11], [12], [13], [14]. To apply this tool, we first establish the formulation for the computation of two parameters, workload and interference of virtual links in the presence of contentions at every network node. We then derive an upper bound of end-to-end delay represented by the worst case response time. One of the advantages of this approach is that it also allows building criteria for schedulability assessment, which is a critical test in order to ensure the deterministic guarantees of the network. Numerical analysis shows that the formulation of RTA leads to a tighter end-to-end delay bound than the one obtained by using Network Calculus. Note that the computation of delay bound will provide hints about how to improve the performance of the considered system. Based on this idea, the approach adopted in the present work is to find the optimal scheduling scheme among a set of scheduling policies, which would lead to the tightest worst-case end-to-end delay bound for a
network modeled by leaky-bucket constrained traffics and rate-latency services for data flow aggregation. It can be shown that in such a system, the optimal scheduling policy will be the one that leads to the minimum burst at the output of destination ES.

The remainder of the paper is organized as follows. Section II presents the adopted AFDX network model. A response-time based end-to-end delay bound analysis and a schedulability analysis are derived in Section III. Section IV addresses the issue of optimizing the scheduling policy in order to obtain the tightest output arrival curve. A case study is carried out in order to validate the proposed approach. Simulation results are reported in Section V followed by some concluding remarks in Section VI.

II. AFDX NETWORK MODELING

AFDX uses the concept of virtual link to identify data flows. A virtual link is defined as a logical unidirectional connection from one source End System (ES) to one or more destination ES according to static routing tables in switches.

The maximum bandwidth allocated to a particular virtual link \( vl_i \) is determined by two parameters:

1) the Bandwidth Allocation Gap (BAG), which is the minimum transmission time interval between the starting bits of two adjacent frames, provided the jitter is zero, whose value is \( 2^n(\text{ms}), n = 0,1,\ldots,7; \)

2) the maximum frame size, denoted by \( l_i^{\text{max}} \), which is a configurable parameter ranging from 84 bytes to 1538 bytes.

Each ES contains single or multiple logical queues and a single server. In this context, the server will be shared based on a given scheduling policy. Note that based on the AFDX standard, each virtual link can aggregate up to four sub-virtual links using the Round-Robin scheduling algorithm [1]. The aggregation of virtual links from sub-virtual links is application specific and not tunable.

A traffic shaper is associated to each virtual link, which is responsible of controlling the bandwidth allocated in such a way that the flow of the corresponding virtual link sends no more than one packet in each BAG interval. One of the traffic regulation algorithms recommended by the AFDX standard is the well known leaky bucket algorithm [1]. A leaky bucket can be characterized by the shaping curve \( \alpha_{\sigma, \rho,t} \) defined as [6]:

\[
\alpha_{\sigma, \rho,t} = \rho t + \sigma.
\]

For a \( vl_i \) with \( BAG_i \) and \( l_i^{\text{max}}, \) the regulator shapes the output flow to satisfy the \((\sigma_i, \rho_i)\)-constraint, where \( \sigma_i = l_i^{\text{max}} \) and \( \rho_i = l_i^{\text{max}}/BAG_i \). The flow of \( vl_i \) will then be constrained by an arrival curve:

\[
\alpha_i(t) = \rho_i t + \sigma_i. \tag{1}
\]

The multiplexer (MUX) can be characterized by the service curve. In this work, a Rate-Latency function is used to describe the service curve provided by the multiplexer on an ES or a switch (SW). Such a function is of the following form:

\[
\beta(t) = C(t - T), \tag{2}
\]

where \( C \) is the service rate of physical link and \( T \) is the latency experienced in the corresponding network node. This curve provides an effective way in the analysis of various services offered by an ES or a SW in the worst case.

The service offered by a multiplexer to each virtual link depends on the adopted scheduling policy. The service curve on a non-preemptive node serving two flows with static priority has been derived in [6] (Section 6.2.1). This result can be extended to a more generic case. More specifically, assume that the node guarantees a strict service curve \( \beta(t) \) to the aggregate flow based on static priority. Furthermore, assume that flows are ordered in such a way that the priority of flow-\( i \) is lower than that of flow-\( j \) if \( i > j \). The service offered by the node to flow-\( i \) is then given by:

\[
\beta_i(t) = \sup \left[ 0, \beta(t) - \sum_{j=0}^{i-1} \alpha_j(t) - \max_{j>i}\{\sigma_j}\right], \tag{3}
\]

where \( \alpha_j(t) \) and \( \sigma_j \) are, respectively, the arrival curve and the burst of flow-\( j \). (3) can be derived by iterating the formula for two flows with static priority while taking into account the fact that on a non-preemptive node, a higher priority flow may also be blocked by at most one flow with lower priority.

In AFDX networks, each output port of switch can be seen as an ES for which the data transmission is controlled by a MUX. For simplicity, delays having a fixed value independent of the scheduling policy, such as forwarding latency, are not considered in the following analysis. Hence, a switch can be represented by the corresponding multiplexers through which the virtual links pass.

The packets in a virtual link \( vl_i \), after entering the MUX, are buffered into a FIFO queue, denoted by \( Q_i \). The scheduler selects buffered data from a set of queues according to the scheduling algorithm. If
the scheduler sends data based on priority, deadline, or another property of the virtual link, each FIFO queue would only receive a subset of the virtual links having a given property. The service provided to queue \( Q_i \) can be described by a rate-latency curve \( \beta_{Q_i} = C_Q \times (t - T_Q) \), where \( C_Q \) is the output rate that the MUX provides to the queue \( Q_i \) and \( T_Q \) is the maximum latency that queue \( Q_i \) waits to be scheduled to export data in a busy period.

III. END TO END DELAY ANALYSIS

This section is dedicated to the development of an RTA-based end-to-end delay analysis. For the purpose of comparison, we first summarize a NC-based formulation of the end-to-end delay bound presented in [15].

A. Analysis Based on Network Calculus

In the framework of NC analysis, the delay of a flow on a node can be represented by the maximum horizontal distance between the arrival and service curves [6]. In case where the virtual link \( vl_i \) is \( (\sigma_i, \rho_i) \)-constrained and the offered service is described by Rate-Latency function given in (4), according to (5), the service offered to \( Q_i \) is [15]:

\[
\beta_{Q_i}(t) = \left( C - \sum_{j=1}^{i-1} \rho_j \right) \left( t - \frac{\sum_{j=1}^{i-1} \sigma_j + \max_{j>i} \{ \sigma_j \}}{C - \sum_{j=1}^{i-1} \rho_j} \right).
\]

(4)

Hence, the service rate and the latency offered to \( Q_i \) are given by:

\[
C_{Q_i} = C - \sum_{j=1}^{i-1} \rho_j,
\]

(5a)

\[
T_{Q_i} = \frac{\sum_{j=1}^{i-1} \sigma_j + \max_{j>i} \{ \sigma_j \}}{C_i}.
\]

(5b)

Therefore, the service curve offered to \( vl_i \) in the queue \( Q_i \) is given by:

\[
\beta_i = C_i \times (t - T_i),
\]

(6)

where \( C_i \) is the output rate allocated to \( vl_i \) by MUX. Consequently, the service rate and the latency offered to \( vl_i \) are:

\[
C_i = C_{Q_i} - \rho_i' = C_{Q_i} - \sum_{vl_j \in Q_i, vl_j \neq vl_i} \rho_j,
\]

(7a)

\[
T_i = T_{Q_i} + \frac{\sigma_i'_{Q_i}}{C_{Q_i}} = T_{Q_i} + \frac{\sum_{vl_j \in Q_i, vl_j \neq vl_i} \sigma_j}{C_{Q_i}},
\]

(7b)

where \( T_i \) is the total scheduling preparation time of \( vl_i \), consisting of two parts. The first part is the inherent delay \( T_{Q_i} \) generated by the scheduling algorithm and the other one is the latency \( \sigma_i'_{Q_i}/C_{Q_i} \) arising from service in the FIFO queue.

Assuming that \( vl_i \) passes \( m_i \) switches across the network, the end-to-end service curve offered to \( vl_i \) is given by the convolution of the service curve of every network node [6]:

\[
\beta_{i,e2e} = \beta_{i,0} \ominus \beta_{i,s1} \ominus \beta_{i,s2} \ldots \ominus \beta_{i,sm},
\]

(8)

where \( \beta_{i,sl} \) is the service curve provided by the source ES (for \( i = 0 \)) or the \( i \)th switch (for \( i > 0 \)).

Denoting the maximum buffering time of \( vl_i \) by \( T_{i}^{\max} = l_{i}^{\max}/C \), where \( l_{i}^{\max} \) is the maximum packet length in \( vl_i \), an end-to-end delay bound is given by [15]:

\[
D_{i,e2e} = \frac{\sigma_i}{C_{i,e}} + T_{i,e} = \min_{0 \leq j \leq m} \{ C_{i,j} \} + m_i T_{i}^{\max} + \sum_{j=0}^{m_i} T_{i,sj}.
\]

(9)

Furthermore, when an arrival curve \( \alpha_i \) passes by a network providing a service curve \( \beta_{i,e2e} \), the corresponding output flow is constrained by an arrival curve \( \alpha_i^*(t) = \alpha_i(t) \ominus \beta_{i,e2e}(t) \), where \( \ominus \) is the min-plus deconvolution [6]. This means that if \( \alpha_i(t) = \rho_i t + \sigma_i \), \( \beta_{i,e2e}(t) = C_{i,e} (t - T_{i,e}) \), and \( \rho_i < C_{i,e} \), then

\[
\alpha_i^*(t) = \rho_i T_{i,e} + \sigma_i,
\]

(10)

where

\[
\sigma_i^* = \rho_i T_{i,e} + \sigma_i
\]

(11)

representing the burst of the output flow.

The aggregate output arrival curve is then given by:

\[
\alpha^* = \sum_{i=1}^{N} \alpha_i^*(t),
\]

(12)

where \( N \) is the total number of virtual links received by the appropriate destination ES.

It is worth noting that Rate-Latency multiplexing affects only the burst of the output arrival curve, but not its rate. Therefore, the optimal arrival curve leading to the minimum delay should be the one with minimum burst. This is an important property which allows simplification of scheduling algorithms design and analysis.
B. Response-Time Analysis

Similar to the definition in [16], the response time of a given virtual link $vl_i$, denoted by $R_i$, is the maximal transmission time of a given packet from the source ES to the destination one. The RTA amounts then to computing the worst-case transmission delay of all packets within a given deadline, which is the BAG in the case of AFDX networks. Therefore, the worst-case response time (WCRT) represents an upper bound of end-to-end delay.

The RTA is an effective technique that has been widely used to derive schedulability tests and other properties for various real-time computing systems [17], [10], [11], [16], although it has initially been developed for fixed priority scheduling [10], [16]. Basically, the RTA lies on the concepts of critical instant and busy period. A critical instant of a task is an arrival time of an instance that suffers from the worst possible interference. This concept allows finding the worst possible response time for the considered task. For fixed priority scheduling of single processor, the simultaneous activation of all tasks represents a critical instant [18]. A schedulability test for periodic and sporadic task sets can then be derived by checking the response time of all tasks in an interval starting with a critical instant in which jobs are released as soon as possible, and comparing it to the corresponding deadlines. While applying the RTA to the worst-case transmission delay analysis of AFDX networks, a multiplexer can be treated as a processor in real-time computing systems. Therefore, the work on this subject can be greatly inspired from a very rich existing literature.

Obviously, when considering only the contention at the level of ES in AFDX networks using non-preemptive static-priority scheduling, simultaneous arrival of packets in all virtual links represents the worst-case scenario. A virtual link $vl_i$ will be blocked by all others with higher or equal priority and by at most one of those belonging to the set of lower priority packets $lp(i)$, for which the maximum blocking time is denoted by

$$B_i = \max_{vl_j \in lp(i)} \{ T_j^{\text{max}} \}. \quad (13)$$

The term $T_j^{\text{max}}$ that corresponds to the transmission time of the packet having the maximum size, is given by:

$$T_j^{\text{max}} = \frac{T_j^{\text{max}}}{C} \quad (14)$$

where $C$ is the service rate of a physical link and $T_j^{\text{max}}$ is the maximum packet size.

To determine the WCRT $R_i$ for $vl_i$ with a priority level $i$, we can directly apply the notion of level-$i$ busy period [19] and $R_i$ is given by the minimal solution of the following recursive equation

$$R_i = T_i^{\text{max}} + B_i + \sum_{vl_j \in hp(i)} \left( R_i \frac{R_i}{BAG_j} \right) T_j^{\text{max}}. \quad (15)$$

The solution of (15) can be computed iteratively, starting at $R_i = T_i^{\text{max}} + B_i$.

Similar to the case of multiprocessor systems, in an AFDX network with multiple end systems and switches, the simultaneous arrival of packets in all virtual links at every source ES may not be the worst-case. To effectively use the concept of interference in RTA, we need to derive the workload of $vl_i$ in a window $[r_i, r_i + R_i)$, where $r_i$ is the arrival time of a packet in $vl_i$ at the level of ES. If more than one packet of $vl_j$ is transmitted during a busy period of $vl_i$, it is necessary to determine the workload of $vl_j$ in order to find the overall interference of $vl_j$ on $vl_i$ [14], [20]. It is straightforward that in a non-preemptive mode, the workload of $vl_j$ during a level-$i$ busy period of length $L$ is bounded by

$$W_j(L) = \left[ \frac{L}{BAG_j} \right] T_j^{\text{max}} + \min (L \mod BAG_j, T_j^{\text{max}}). \quad (16)$$

Note that for any $L \geq m_i T_i^{\text{max}}$, $vl_j$ cannot contribute to the interference on $vl_i$ for more than $L - m_i T_i^{\text{max}}$. Therefore, the interference of $vl_j$ on $vl_i$ is given by

$$l_j(L) = \min (W_j(L), L - m_i T_i^{\text{max}}) \quad (17)$$

To simplify the analysis, we further suppose that any two virtual links can intercept at most one time no matter how long is their common path. In this case, we can employ the principle of “paying for multiplexing only once” [21], which states that a virtual link can be blocked by another one only the first time when they intercept. In other words, once there is a contention between two virtual links, the virtual link with lower priority is delayed once for all by the one with higher priority. This delay will create an adjustment or a kind of synchronization between the two corresponding virtual links such that the delay does not happen a second time in the newly formed flow. Similarly, the maximum blocking time that $vl_i$ experiences due to lower-priority virtual links can be expressed as:

$$B_i^{\text{m}} = \sum_{n=0}^{m_i} \max_{vl_j \in lp_n(i)} \{ T_j^{\text{max}} \}, \quad (18)$$
where $lp_{n}(i)$ is the set of lower priority virtual links at the level of $n$th node. By using the same iterative method that is used for multiprocessor scheduling [20], $R_i$ is given by the minimal solution of the following recursive equation:

$$R_i = m_i T_i^{\text{max}} + B_i^{m_i} + \sum_{\forall v l_j \in hp(i)} I_j(R_i),$$

(19)

where $hp(i)$ is the set of all virtual links with priority higher than or equal to $vl_i$ encountered by $vl_i$ at the level of every network node and $I_j(R_i)$ is the interference of $vl_j$ on $vl_i$ during the time interval $R_i$. The solution of (19) can be found by an iteration starting at $R_i = m_i T_i^{\text{max}} + B_i^{m_i}$.

Schedulability criteria can also be derived from RTA. In fact, whatever the adopted scheduling policy, the end-to-end delay of each virtual link must be bounded by the corresponding BAG. Consider a set of input $vl_i$, $i = 1, \ldots, N$, sharing common paths. In a way similar to [22], [23], a schedulability test can be expressed as:

$$m_i T_i^{\text{max}} + B_i^{m_i} + \sum_{\forall v l_j \in hp(i)} I_j(BAG_i) \leq BAG_i,$$

(20)

for $i = 1, \ldots, N$. Conceptually, the schedulability condition given in (20) implies that the end-to-end transmission of a packet in $vl_i$ should be finished within the corresponding $BAG_i$ even in the worst case. This shows the sufficiency of the condition given in (20).

### IV. Optimization of Output Arrival Curves

As mentioned in Section III-A, the end-to-end latency depends only on the burst of the aggregate output flow. Therefore, the optimal scenario among all possible static priority scheduling schemes can be obtained by

$$s_{\text{opt}} = \arg \min_{s \in \text{SP}} \left\{ \sum_{i=1}^{N} T_{i,e}^s \rho_i + \sum_{i=1}^{N} \sigma_i \right\}$$

$$= \arg \min_{s \in \text{SP}} \left\{ \sum_{i=1}^{N} T_{i,e}^s \right\},$$

(21)

where $\text{SP}$ is a set of all possible static-priority scheduling scenarios on the AFDX network and $T_{i,e}^s$ is the end-to-end latency encountered by $vl_i$ under a particular scenario $s$. Note that in this optimization problem, $T_{i,e}^s$ is the only term depending on the scheduling policy. Therefore, the scheduling policy leading to the tightest arrival curve is the one that offers the minimal accumulated latency bound.

Furthermore, $T_{i,e}^s$ can be computed by either (21), based on Network Calculus, which reads:

$$T_{i,e}^s = \sum_{j=0}^{m_i} T_{i,sj}$$

(22)

or by (15), based on response time analysis, which is given by:

$$T_{i,e}^s = R_i - m_i T_i^{\text{min}},$$

(23)

where $T_i^{\text{min}} := \frac{t_i^{\text{min}}}{C}$.

In a static scheduling scheme, the priority can be assigned based on different considerations. As virtual links are characterized by their BAG and rate, we are particularly interested in scheduling policies related to two static priority assignment schemes, namely BAG-based and rate-based scheduling. In the BAG-based scheduling scheme, higher priority is assigned to virtual links with smaller BAGs. As in AFDX networks it is assumed that the packets of a given virtual link are transmitted periodically with a time interval defined by the BAG, this scheme is indeed an implementation of the well known Rate-Monotonic Algorithm (RMA) [18], [24], [25]. Another way to provide a deterministic data transmission is to guarantee the maximum throughput for every virtual link. This leads to a scheduling scheme that assigns higher priority to virtual links with higher transmission rate.

It is shown in [18] that in the case of single processor systems, the RM scheduling is optimal among all static-priority scheduling policies when the deadline of each task is the same as its period and the optimality holds for both preemptive and non-preemptive schemes. Therefore, for a single network node, BAG-based scheduling is optimal in the sense defined in [18] and should lead to the tightest output arrival curve. However, as in the case of multiprocessor, this optimality may not hold for multi-hop links in AFDX networks. Nevertheless, we can always use the optimization procedure proposed in (21) to find the scheduling policy that experiences the least accumulated end-to-end delay. Note that for a system with fixed routing topology, one can always find the optimum of (21) by permuting the priority assigned to each virtual link. However, for a large network, direct permutation may lead to excessive computational complexity. In this case, there is a need to look into more efficient methods to solve the problem given in (21).

To illustrate the influence of different static-priority schemes to the arrival curve of output flow and end-to-end delay, we consider an AFDX network shown in...
This network is composed of three interconnected switches and seven end systems. The specific configuration of $vl_1 \sim vl_7$ are shown in Table I. It is assumed that the data transmission rate of the physical link is 10 Mbps. The priority of each virtual link corresponding to BAG-based and rate-based scheduling is also shown in Table I. For this configuration, there are in total $7!$ possible scheduling scenarios.

![AFDX Network Configuration](image)

**Fig. 1. AFDX Network Configuration.**

<table>
<thead>
<tr>
<th>VL</th>
<th>BAG (µs)</th>
<th>$l_{max}$ (bytes)</th>
<th>BAG-based priority</th>
<th>Rate-based priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>$vl_1, vl_4$</td>
<td>4000</td>
<td>120</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$vl_2, vl_5$</td>
<td>16000</td>
<td>320</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$vl_3, vl_6$</td>
<td>32000</td>
<td>1000</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>$vl_7$</td>
<td>2000</td>
<td>120</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Fig. 2. TrueTime Simulation Model.**

To represent a large number of packet overlap scenarios, the transmission starting time of each $vl_i$ is chosen randomly between 0 to 1 ms. In addition, the packet size of each $vl_i$ is set to $l_{i}^{max}$, which leads to the maximum transmission delay. In reported experiments, 10,000 packets were generated for each virtual link in order to simulate realistic scenarios. In the simulations, the same scheduling scheme is applied to all the multiplexers at the level of source end-systems and switches.

**TABLE II**

<table>
<thead>
<tr>
<th>VL</th>
<th>BAG-based (RTA)</th>
<th>BAG-based (NC)</th>
<th>Rate-based (RTA)</th>
<th>Rate-based (NC)</th>
<th>FCFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>937</td>
<td>1108</td>
<td>990</td>
<td>1084</td>
<td>1125</td>
</tr>
<tr>
<td>$S_2$</td>
<td>937</td>
<td>1108</td>
<td>990</td>
<td>1084</td>
<td>1125</td>
</tr>
<tr>
<td>$S_3$ (ES6)</td>
<td>2774</td>
<td>2988</td>
<td>2726</td>
<td>3064</td>
<td>3928</td>
</tr>
<tr>
<td>$S_3$ (ES7)</td>
<td>752</td>
<td>882</td>
<td>874</td>
<td>1018</td>
<td>1052</td>
</tr>
</tbody>
</table>

In order to evaluate the RTA-based end-to-end delay bound estimation algorithm, simulation studies are carried out to measure the delay encountered by $vl$s in the AFDX network shown in Fig. 1. Again, for the purpose of comparison, three simple scheduling policies, namely BAG-based, rate-based, and FCFS schemes, are considered with the configuration specified in Table I. The platform used in simulation is a real-time network control system simulation software called TrueTime [26]. Figure 2 illustrates the TrueTime platform used to simulate our AFDX network where each $vl$ is specified by the corresponding BAG and $l_{max}$.
from Fig. 3 that among the considered schemes, the algorithm FCFS is the worst one, which leads to the largest total delay. Whereas, it is clear that BAG-based scheduling encounters the smallest total delay.

Figure 4 shows the estimated end-to-end delay bounds obtained by NC and RTA, as well as the maximum delay measured by simulation. It can be seen that the estimates based on RTA are closer to those obtained by simulation for both BAG-based and rate-based scheduling. Obviously, in the case of FCFS the estimated delay bounds obtained by RTA and NC are identical, because no priority order is taken into account. We can also note that the estimated delay bound in the case of FCFS is very pessimistic due to the unknown scheduling order. On the other hand, there exists a gap between the estimated delay bounds and the simulated values, in particular in the case of FCFS.

VI. CONCLUSION AND FUTURE WORK

This work addresses the applicability of response time analysis to the upper bound computation of the end-to-end delay in AFDX networks. We establish firstly the formulation for both the workload and interference computation of virtual links in the presence of contentsions at the network nodes they go through. We then derived the end-to-end delay upper bound described by the worst case end-to-end response time. Delay bound analysis leads also to an optimization procedure that allow finding the optimal scheduling policy with minimum end-to-end delay. Numerical analysis and simulation study confirmed the effectiveness of the proposed method and the tightness of the estimated delay bound compared to that obtained by Network Calculus. The results also show the advantage of BAG-based scheduling over rate-based and FCFS scheduling policies. It was observed that BAG based scheduling is an implementation of the rate monotonic scheduling in the context of AFDX networks. BAG based scheduling led to the best obtained solutions, although it may not be optimal for multi-hop networks. This paper shows that good scheduling policies can be obtained by numerical analysis.

ACKNOWLEDGMENTS

This work is funded by NSERC-CRIAQ CRD project AVIO402, MITACS Acceleration Quebec program, MDEIE/CRIAQ Quebec-China project, and sponsored by Thales Canada Inc. and Bombardier Aerospace.

REFERENCES


L. George and D. De Voluceau, “Preemptive and non-preemptive real-time uni-processor scheduling,” 1996.


