

A probabilistic analysis of end-to-end delays on an AFDX avionic network

Jean-Luc Scharbag, Frédéric Ridouard, Christian Fraboul
Université de Toulouse - IRIT/ENSEEIH/INPT
2, rue Camichel - 31000 Toulouse - France
{Jean-Luc.Scharbag,Frédéric.Ridouard,Christian.Fraboul}@enseeiht.fr

Abstract

AFDX (Avionics Full Duplex Switched Ethernet, ARINC 664) developed for the Airbus A380 represents a major upgrade in both bandwidth and capability. Its reliance on Ethernet technology helps to lower some implementation costs, but guaranteed service presents challenges for system designers.

An analysis of end-to-end transfer delays through the network is required in order to determine upper bounds. In this paper, we propose to compute probabilistic upper bounds for end-to-end delays on avionic flows. Such upper bounds can be exceeded with a given probability p , and are relevant in the context of avionics, where functions are designed to give accurate results even if they miss some frames.

The stochastic network calculus approach analytically determines a probabilistic upper bound, whereas the simulation approach gives an experimental upper bound. The former may be used for new certification needs since it assures that the probability of exceeding the computed upper bound is not greater than p . The latter closely approximates actual network behavior and can help to give some idea of the pessimism of the stochastic network calculus upper bound.

The two approaches have been developed in the context of an industrial AFDX network configuration.

1 Introduction

Thanks to the Integrated Modular Avionics concept [1, 2], functions developed for civilian aircraft share computation resources. However, the continual growing number of these functions implies a huge increase in the quantity of data exchanged and thus in the number of connections between functions. Consequently, traditional ARINC 429 buses [3] can't cope with the communication needs of modern aircraft. Indeed, ARINC 429 is a single-emitter bus with limited bandwidth and a huge number of buses would be required. Clearly, this is unacceptable in terms of weight and complexity.

In order to cope with this problem, the AFDX (Avionics Full Duplex Switched Ethernet) [4, 5, 6] was defined

and has become the reference communication technology in the context of avionics. AFDX is a full duplex switched Ethernet network to which new mechanisms have been added in order to guarantee the determinism of avionic communications. This determinism has to be proved for certification reasons and an important challenge is to demonstrate that an upper bound can be determined for end-to-end communication delays.

An important assumption is that all the avionics communication needs can be statically described: asynchronous multicast communication flows are identified and quantified. All these flows can be statically mapped on the network of AFDX switches. For a given flow, the end-to-end communication delay of a frame can be described as the sum of transmission delays on links and latencies in switches. Thanks to full duplex links characteristics, no collision can occur on links [12] and transmission delays on links depend solely on bandwidth and frame length. But, as confluent asynchronous flows compete, on each switch output port (according to a servicing policy), highly variable latencies can occur when a frame crosses a switch. Thus it is necessary to analyze these latencies in order to determine the upper bounds on end-to-end communication delays for each flow.

The first step, mainly for avionic network certification purpose, was to use the deterministic network calculus theory in order to compute a worst-case upper bound for each communication flow of the avionic applications on an industrial AFDX network configuration [10]. This worst-case communication delay analysis allowed the comparison between the computed upper bounds and the constraints on the communication delays of each flow. Moreover it allowed the scaling of the switches memory buffers in order to avoid buffer overflow and frame losses. But such a worst case communication analysis is obviously pessimistic. Indeed, communication delays measured on a real configuration are much lower than the computed upper bound. This is mainly due to the fact that network calculus theory makes pessimistic assumptions on simultaneously arriving flows. It is also due to the fact that rare events are difficult to observe on a real configuration in a reasonable time.

In order to better understand the real behavior of the AFDX network, a simulation model of the network is pro-

posed as a second step. Such a simulation approach allows the calculation, on the modeled network, of the end-to-end delay for each flow, according to a representative subset of possible scenarios. Thus an end-to-end delay distribution can be obtained for each flow, leading to a better understanding of communication delays. However such an approach cannot be used for certification needs as rare events can be missed by simulation.

In a third step, stochastic network calculus theory is proposed to compute a probabilistic upper bound. This theory allows the computation of the probability p for an end to end delay to exceed a given bound. This probability p can be interpreted as the acceptable probability that a frame misses its deadline. Such a result could be useful for new certification needs as many avionic functions are designed to give accurate results even if they miss some frames.

This paper focuses on the probabilistic analysis of end-to-end delays on an avionics AFDX network. It considers both the simulation and the stochastic network calculus approaches. It shows how these two approaches are applied in an industrial application context.

The paper is organized as follows. Section 2 details the main objectives of the study. Section 3 presents the simulation approach. Specifically, it shows how the simulation space can be drastically reduced by focusing on the part of the network which influences the end-to-end delay of a given flow. Section 4 presents the stochastic network calculus approach. Examples of end-to-end delay analysis on an industrial network are presented in section 5. Section 6 concludes and indicates directions for future research.

2 Main objectives of the study

This section presents the main challenges of applying a probabilistic analysis of end-to-end delays on an avionics switched Ethernet network, the characteristics of which are briefly summarized. Two complementary approaches are introduced, i.e. a simulation approach and a stochastic network calculus one.

2.1 The industrial AFDX network context

The AFDX is a switched Ethernet network taking into account avionic constraints. Figure 1 depicts an illustrative example. It is composed of five interconnected switches $S1$ to $S5$. There are no buffers on input ports and there is one FIFO buffer for each output port. The inputs and outputs of the network are called *end systems* ($e1$ to $e10$ in figure 1). Each end system is connected to exactly one switch port and each switch port is connected to at most one end system. Links between switches are all full duplex.

The end-to-end traffic characterization is made by the definition of Virtual Links. As standardized by ARINC-664, Virtual Link (VL) is a concept of virtual communication channels. Thus it is possible to statically define the flows which enter the network [6].

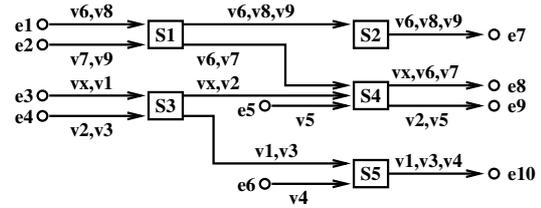


Figure 1. AFDX network architecture

End systems exchange Ethernet frames through VLs. Switching a frame from a transmitting to a receiving end system is based on a VL. The Virtual Link defines a logical unidirectional connection from one source end system to one or more destination end systems. Coming back to the example in figure 1, vx is a unicast VL with path $e3 - S3 - S4 - e8$, while $v6$ is a multicast VL with paths $e1 - S1 - S2 - e7$ and $e1 - S1 - S4 - e8$.

The routing of each VL is statically defined. Only one end system within the avionics network can be the source for each Virtual Link, (i.e., Mono Transmitter assumption). A VL definition also includes the Bandwidth Allocation Gap (BAG) and the minimum and the maximum frame lengths (s_{min} and s_{max}). BAG is the minimum delay between two consecutive frames of the associated VL (which actually defines a VL as a sporadic flow).

The parameters of each VL (BAG , s_{max}) are assured by a shaping unit added on the corresponding emitting end system and a policing unit added on the first switch input port crossed by the VL (it is the only specificity of AFDX switches, compared with standard Ethernet switches).

Typically, an industrial AFDX network includes more than one hundred end systems and two redundant AFDX sub-networks, each composed of eight switches. Nearly 1000 Virtual Links are transmitted on each sub-network, corresponding to more than 6000 paths due to the multicast characteristic of VLs.

2.2 The modeling and simulation approach

The goal of the simulation approach is to approximate real network behavior. This approach needs a realistic model of the network and calculates the end-to-end delay of a given flow on a subset of all possible scenarios. Thus, the end-to-end delay distribution of that flow can be obtained, provided the considered subset is representative of all possible scenarios.

Section 3.1 shows that an industrial network leads to a huge number of possible scenarios. Consequently, finding a representative subset of scenarios in order to calculate the end-to-end delay distribution of a given flow is not easy. The key idea proposed in this paper is to model only the elements of the network configuration (VLs, output ports, links) which have an influence on the end-to-end delay distribution of the flow. These elements constitute the part of the network which is relevant to the flow.

2.3 The stochastic network calculus approach

As mentioned in the introduction, certification is mandatory in the context of avionics. This cannot be obtained without a safe probabilistic upper bound on the end-to-end delay of each flow. An exact stochastic analysis of an industrial avionic network is unaffordable, due to the number of VLs of such a network configuration. One way to solve this problem is to use a pessimistic stochastic analysis which is a safe approximation of the exact stochastic analysis. This concept of pessimistic analysis is introduced in [15]. The pessimistic analysis is a safe approximation in the sense that the probability of exceeding the end-to-end delay bound it provides is guaranteed to be greater than the exact one. In other words, the calculated upper bound associated with a given probability is guaranteed to be greater than the exact upper bound.

The simulation approach presented in the previous section gives an approximation of the end-to-end delay distribution which leads to an experimental upper bound on end-to-end delays. This upper bound can be either optimistic or pessimistic. Thus, this experimental upper bound is not safe and this approach cannot be used for certification in the context of avionics.

The stochastic network calculus approach is based on the same modeling assumptions as the simulation approach. It can analytically determine a probabilistic upper bound on the end-to-end delay of a given flow mapped on a given network, provided a set of properties are verified. This approach is a pessimistic analysis, since it is based on pessimistic assumptions. Consequently, the stochastic network calculus approach could be a good candidate for new certification needs. In section 4 the required properties of the stochastic network calculus are verified in the context of an AFDX configuration.

3 End-to-end delay analysis through a simulation approach

Before presenting the simulation approach, an overview is given on the different parts of a frame end-to-end delay on an AFDX network.

Let's consider a VL path p_x . The end-to-end delay D_{Fp_x} of a frame Fp_x transmitted on p_x is defined by

$$D_{Fp_x} = LD_{Fp_x} + SD_{Fp_x} + WD_{Fp_x}$$

where:

- LD_{Fp_x} is the transmission delay over the links: thanks to the full duplex characteristic of AFDX, there are no collisions on the links. Thus, the transmission delay over a link is $c \times s_{Fp_x}$ where c is the link bandwidth and s_{Fp_x} is Fp_x length. Therefore, considering that all the links have the same bandwidth c ,

$$LD_{Fp_x} = nbl_{p_x} \times (c \times s_{Fp_x})$$

where nbl_{p_x} is the number of links in p_x .

- SD_{Fp_x} is the delay in switches between input and output ports: in the context of this paper, the delay in a switch from an input port to an output port is considered as a constant SD , since the only available information about this delay is a guaranteed upper bound of $16 \mu s$. Thus

$$SD_{Fp_x} = nbs_{p_x} \times SD$$

where nbs_{p_x} is the number of switches in p_x .

- WD_{Fp_x} is the delay in switches and end system output buffers: this delay highly depends on each output port load at the time where Fp_x reaches it, as will be illustrated in section 3.1. Thus

$$WD_{Fp_x} = WD_{Fp_x}(ES_{p_x}) + \sum_{Sk \in \Psi_{p_x}} WD_{Fp_x}(Sk)$$

where ES_{p_x} is Fp_x source end system, Ψ_{p_x} is the set of switches in p_x , $WD_{Fp_x}(ES_{p_x})$ is the delay in ES_{p_x} output buffer and $WD_{Fp_x}(Sk)$ is the delay in Sk output port buffer.

Consequently, D_{Fp_x} can be divided into a fixed part $LD_{Fp_x} + SD_{Fp_x}$ and a variable part WD_{Fp_x} . The fixed part can be statically computed since it depends solely on the path p_x , Fp_x length and links bandwidth. The variable part depends on the scenario that is defined in the next section.

3.1 Simulation scenario parameters

A simulation scenario is defined by considering, on the one hand, the VL characteristics and, on the other hand, the interferences between VLs.

As presented in section 2, a VL is defined by the minimum delay between the emission of two consecutive frames (the BAG) and the minimum and the maximum frame lengths s_{min} and s_{max} . A VL Vi transmits a given set of application data Vi_{data_i} . At the beginning of each BAG, a subset of that application data is ready for transmission. This subset can be \emptyset , Vi_{data_i} or a predefined non empty part of Vi_{data_i} . If \emptyset is ready, no frame is transmitted. Vi_{data_i} defines s_{max} while the smallest predefined non empty part of Vi_{data_i} defines s_{min} . Thus, at the beginning of each BAG, each VL transmits either no frame or a frame with a length between s_{min} and s_{max} . Therefore, the delay between two consecutive frames of a VL is a multiple of its BAG (a VL is a periodic flow with holes). An example of frame emissions for a VL is depicted in figure 2. In this example, there are four possible frame

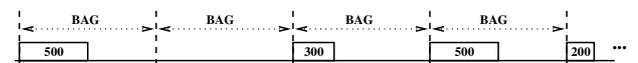


Figure 2. Emissions of frames for a VL

lengths: 0 bytes (\emptyset), 200 bytes (s_{min}), 300 bytes and 500 bytes (s_{max}).

VLs interfere with each other in end systems and switch output port buffers since they share communication links. Obviously, if several frames arrive at the same time at a switch output port, most of them will have to wait to be transmitted. Conversely, if frames arrive at sufficiently spaced intervals at the same output port, all of them will be transmitted immediately. The arrival time of a frame at an output port mainly depends on its emission by the corresponding VL source end system. Thus, interferences between VLs frames are a function of their emission times (the phasing between VLs). Let's consider the example in figure 3, where the link $S1 - e3$ is shared by VLs vx and $v1$. Figure 4 depicts frame transmissions

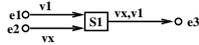


Figure 3. Example of phasing influence

for two possible phasings of vx and $v1$. With phasing (a),

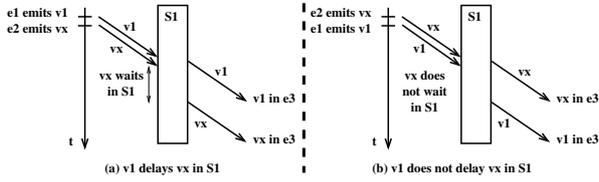


Figure 4. Phasing and waiting time

vx has to wait for the end of transmission of $v1$ frame while it does not have to wait with phasing (b). Moreover, if $v1$ transmits no frame during a given BAG (because it has no data to transmit), the corresponding vx frame will not wait, whatever phasing is considered.

In short, the following parameters define a scenario:

- the sequence of frames emitted by each VL, i.e. BAG occupation and frame lengths,
- the phasing between VLs, i.e. the first frame emission time for each VL. Any phasing is possible, since avionic functions are asynchronous.

It has been previously noted that a typical AFDX network includes approximately 1000 VLs. Clearly, this leads to a huge set of possible scenarios from which it is difficult to extract a representative subset. The resulting challenge is, for each VL path, to focus on the part of the network that is relevant for this path's end-to-end delay distribution in order to reduce the simulation space. This is a mandatory requirement for the simulation approach. It is fulfilled by means of the VLs taxonomy that is presented in the next section.

3.2 A taxonomy of VLs

The basic idea of the taxonomy is that, given a path px of a VL vx , the other VLs do not have the same level of influence on it. For example, a vx frame can wait for the end of transmission of another frame only if the latter shares at least one output port with px . The application of this

idea is to focus the simulation on the VLs that influence the end-to-end delay distribution of vx frames.

The taxonomy is illustrated considering the unicast VL vx in figure 1. Its path px is $e3 - s3 - s4 - e8$.

The paths or portions of paths of other VLs of this AFDX configuration can be divided into three classes [11], as depicted in figure 5.

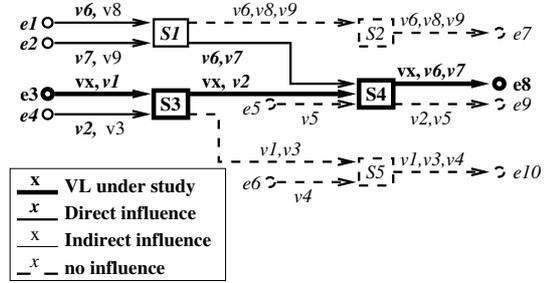


Figure 5. Taxonomy of VLs

- Class *DI* (Direct Influence) contains all the paths that share at least one output buffer with px , truncated after the last output buffer shared with px . In figure 5, it contains the whole VL $v7$, path $e1 - s1 - s4 - e8$ of $v6$ and sub-paths $e3 - s3$ and $e4 - s3 - s4$ of $v1$ and $v2$ respectively.
- Class *II* (Indirect Influence) contains all the paths or portions of paths that share no output buffer with px , but at least one output buffer with a *DI* or an *II* path. In figure 5, sub-paths $e1 - s1$ of $v8$, $e2 - s1$ of $v9$ and $e4 - s3$ of $v3$ are classed as indirect influence portions of VL paths.
- Class *NI* (No influence) contains all the paths or portions of paths that are not in class *DI* or class *II*. It contains all links represented with dashed lines in figure 5.

In this illustrative example containing ten VLs overall, classes *DI* and *II* each contain four and three VLs respectively. Figure 6 shows the partitioning between classes *DI*, *II* and *NI* for each VL path in a sample industrial network including 1000 VLs and 6400 paths. The con-

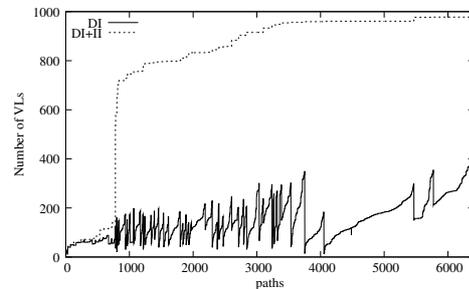


Figure 6. Industrial configuration taxonomy

tinuous and dashed lines respectively give the number of VLs in class *DI* for each path and the number of VLs in classes *DI* or *II*. In this industrial network, on average,

a VL path has 150, 650 and 200 *DI*, *II* and *NI* VLs respectively.

Considering this VL classification, VLs in class *NI* clearly have no impact on the end to end delay of their associated path px . Thus, VLs in class *NI* will not be considered in the definition of a scenario for a px end-to-end delay analysis. For the network analyzed in figure 6, this leads to a drastic reduction of the simulation space for approximately 800 VLs paths (each scenario includes less than 150 VLs instead of nearby 1000). Unfortunately, this reduction is quite poor for the 5600 remaining VLs paths (each scenario includes an average of 800 VLs).

In order to obtain a larger reduction of the simulation space, the VL classification has to be exploited more effectively. The main idea concerns VLs in class *II*. They could be ignored in the definition of a scenario for a px end-to-end delay analysis provided they have no influence on px end-to-end delay distribution. The next section studies the effective influence of VLs in class *II*.

3.3 Effective influence of VLs in class *II*

The influence of a VL in class *II* on px is illustrated the example depicted in figure 7. It includes one switch $s1$,

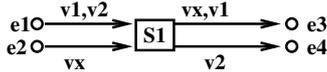


Figure 7. Example of *II* Influence

four end systems $e1, \dots, e4$ and three VLs $vx, v1$ and $v2$. These three VLs have identical BAGs and frame lengths. Using the taxonomy presented in section 3.2, unicast VL vx is directly influenced by $v1$ (class *DI*) and indirectly influenced by $v2$ (class *II*).

Depending on the scenario (phasings for $vx, v1$ and $v2$), $v2$ can have an influence on the vx end-to-end delay by modifying the $v1$ arrival time at the switch $s1$ output port. The three possible cases are illustrated in figure 8, considering three scenarios. For each of them, figure 8 shows the modification of the vx end-to-end delay due to $v2$ frames. For the three scenarios, $v1$ and $v2$ are ready for transmission simultaneously and each $v2$ frame is arbitrarily transmitted before the corresponding $v1$ frame. Thus, the non-transmission of a $v2$ frame advances the arrival time of the corresponding $v1$ frame at the switch $s1$ output port. In scenario a in figure 8, this leads to a shorter vx end-to-end delay because it allows the $v1$ frame to complete transmission on the $s1 - e3$ link before the arrival of the vx frame at the $s1$ output port. Conversely, it leads to a longer vx end-to-end delay in scenario b , because the arrival order of the vx and $v1$ frames at the $s1$ output port is inverted and consequently, the vx frame has to wait. Finally, the non-transmission has no influence in scenario c , because the vx frame arrives before the $v1$ one in both cases and as a result never waits.

Thus, depending on the application scenario, $v2$ frames can shorten, lengthen or have no influence on vx end-to-end delays. However, it remains to be seen if VLs in class

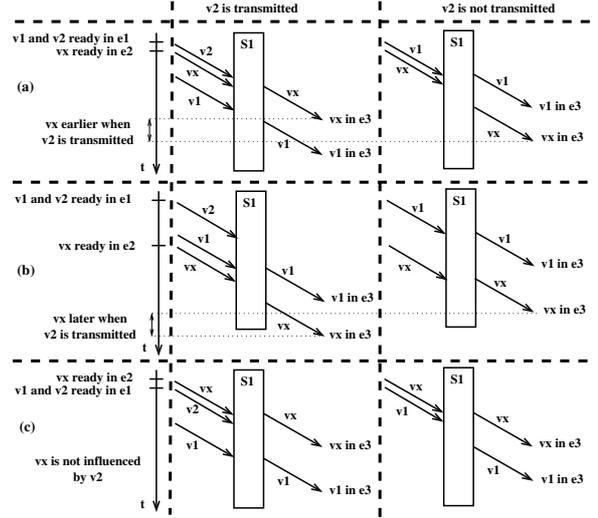


Figure 8. Possible frame arrival times

II (e.g. $v2$) modify the end-to-end delay distribution of px , their associated VL path.

In order to answer this question, every possible VL path must be examined. The basic idea is to determine, for each VL path, the end-to-end delay distributions considering first, that VLs in class *II* are present, and second, that they are not present. The goal is to determine whether VLs in class *II* modify the end-to-end delay distributions (there is at least one VL path for which the two obtained distributions are different) or not (such a VL path does not exist). In the latter case, VLs in class *II* do not have to be taken into account when determining end-to-end delay distributions.

3.3.1 The VL modeling approach

End-to-end delay distributions are obtained using a simulation approach. Such an approach needs a model for each considered VL path. The model corresponding to a given path px includes px , all the VLs in its class *DI* and possibly VLs in its class *II*. The general structure of the model depends on the length of px (number of crossed switches) and on the characteristics of the VLs in classes *DI* and *II*.

Figure 9 shows four generic models that cover all possible VL paths px of length one or two. Each model defines a path px from a VL emitted by end system es and received by end system ed .

- *Model1* covers VL paths of length 1, directly or indirectly influenced by VLs generated by end systems directly linked to switch $s1$ input ports,
- *Model2* covers VL paths of length 2, directly or indirectly influenced by VLs generated by end systems directly linked to switches $s1$ or $s2$ input ports,
- *Model3* is a generalization of *model1*: VLs in classes *DI* and *II* can cross other switches before reaching $s1$,

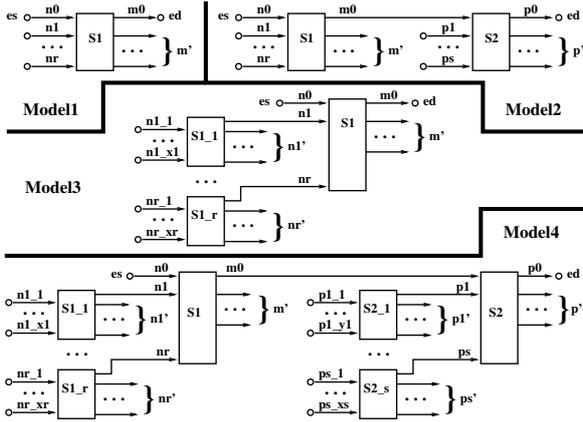


Figure 9. Models of VL path types

- *Model4* is a generalization of *model3*: VLs in classes *DI* and *II* can cross other switches before reaching *s1* or *s2*,

Similar models can be considered for longer paths px .

A more detailed view of *Model1* is depicted in figure 10. Switch *s1* has r input links in addition to link $es - s1$.

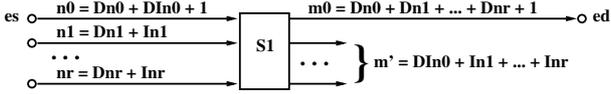


Figure 10. *Model1* detailed view

ni VLs ($0 \leq i \leq r$) are transmitted over the $s1$ i^{th} input link ($es - s1$ is considered as link 0). The $n0$ VLs of link 0 include px , $Dn0$ VLs that are later transmitted on link $s1 - ed$ and $DIn0$ VLs that are then transmitted on another link. The ni VLs ($1 \leq i \leq r$) include Dni VLs that are afterward transmitted on link $s1 - ed$ and Ini VLs that are then transmitted on another link. Thus, class *DI* consists of $Dn0 + DIn0 + Dn1 + \dots + Dnr$ VLs and class *II* is made up of $m' = In1 + \dots + Inr$ VLs, which are transmitted on an arbitrary number of $S1$ output links.

The three other models in figure 9 are constructed in a similar manner. In *Model3*, the $s1$ i^{th} input link ($1 \leq i \leq r$) is an output link of a switch $s1_i$ which has xi input links. The $s1_i$ j^{th} input link ($1 \leq j \leq xi$) conveys ni_j VLs. Among them, Dni_j are transmitted on link $s1_i - s1$ and the Ini_j remaining ones on another $s1_i$ output link. *Model2* and *Model4* are respectively the aggregate of two *Model1* and *Model3* occurrences.

Since the number of possible VL paths is huge, an exhaustive analysis of them all is impracticable in general case. Fortunately, the characteristics of the industrial applications considered in this paper limit the number of possible VL paths. Indeed, a typical industrial AFDX configuration is composed of eight switches. The path length (number of crossed switches) is between one and four and links are lightly loaded. Consequently, a set of VL paths can be defined that covers all the possible VL paths of an industrial network configuration.

The results of the evaluation of the influence of VLs in class *II* were presented in [26]. They are summarized in the following paragraph.

3.3.2 The simulation process

Model1 in figure 9 is considered first. As presented in section 3.3.1, this model is characterized by the number of $s1$ input links, the number of VLs transmitted on each of these input links and the dispatching of these VLs between link $s1 - ed$ and the other $s1$ output links. Moreover, BAGs and frame lengths associated with each VL have to be defined. In order to cover all possible cases in an industrial AFDX network, a typical network was examined and the following range of values deduced:

- between 2 and 8 $s1$ input links,
- between 5 and 118 VLs per $s1$ input links,
- for link $es - s1$, between 1 (vx) and all VLs transmitted on link $s1 - ed$ (the remaining are transmitted on another $s1$ output link),
- for the other $s1$ input links, VLs equally dispatched between link $s1 - ed$ and the other $s1$ output links,
- VLs BAGs among 4, 8, 16, 32, 64 and 128 ms ,
- VLs frame lengths between 84 and 1500 bytes.

More than three hundred *Model1* paths were evaluated. As an illustration, this process is detailed in example $P1$ in figure 11. vx has a BAG of 4 ms and a frame length of 84

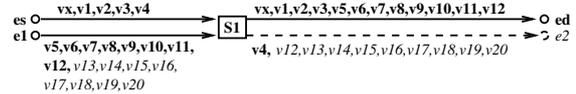


Figure 11. *Model1* example $P1$

bytes ($s_{min} = s_{max}$). It is a *Model1* path with $r = 1$ (two $s1$ input links), $n0 = 5$ (vx and $v1 \dots v4$) and $n1 = 16$ ($v5 \dots v20$). Its class *DI* includes 12 VLs ($v1 \dots v12$) and its class *II* includes 8 VLs ($v13 \dots v19$, in italics in figure 11). All the VLs have the same BAG and frame length as vx . $P1$ is a very lightly loaded configuration. Specifically, link $s1 - ed$ has a load of about 2 % (12 frames of 84 bytes every 4 ms , at 100 Mb/s).

vx end-to-end delay distributions are computed considering first, that VLs in class *II* are transmitted, and second, that they are not transmitted. The same vx end-to-end delay distribution is obtained for both cases. It is depicted in figure 12 ($P1$ curve) using a logarithmic scale. The probability associated with each value of the end-to-end delay is shown with an accuracy of 1 μs . In this example, VLs in class *II* have no influence on vx end-to-end delay distribution. The delay lower bound is 29 μs (vx frames never wait in output buffers). The delay upper bound obtained is 42 μs . This means that no simulated scenario gives a vx end-to-end delay greater than 42 μs .

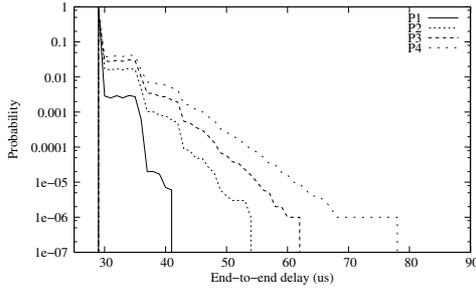


Figure 12. End-to-end delays distributions on *Model1* examples

	r	$n0$	$n1$	$n2$	$Dn0$
$P2$	1	27	88		25
$P3$	2	49	81	81	47
$P4$	2	71	118	118	69

Table 1. *Model1* examples

However, this experimental upper bound is not guaranteed, since simulations can fail to consider rare events.

Figure 12 depicts end-to-end delay distributions for the three other *Model1* examples described in table 1. $P2$, $P3$ and $P4$ correspond to different network loads (respectively 12 %, 22 % and 32 % on link $s1 - ed$).

As for $P1$, whatever case is considered for VLs in class *II*, the same vx end-to-end delay distributions are obtained. These are depicted in figure 12. Not surprisingly, end-to-end delays increase when the network load increases.

Similar evaluations have been done on *Model2*, *Model3* and *Model4* VL paths (more than three hundred instances for each model). No configuration was evaluated where VLs in class *II* influenced the vx end-to-end delays distribution. Figure 13 presents obtained results in a slightly different manner than figure 12. It depicts the probability of exceeding a given end-to-end delay for vx of the *Model2* examples in table 2. All VLs of these six

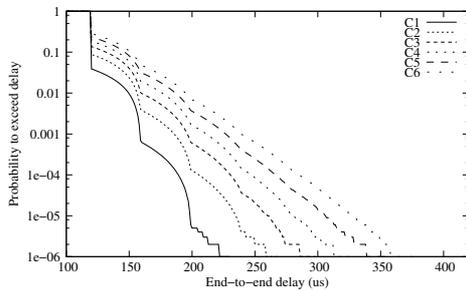


Figure 13. Experimental upper bounds on *Model2* examples

examples have a BAG of 4 ms and a frame length of 500 bytes. The results in figure 13 lead to the same conclusions as those in figure 12.

	r	$n0$	$n1$	s	$m0$	$m1$	$m2$
$C1$	1	1	2	2	2	4	2
$C2$	1	2	4	2	4	8	4
$C3$	1	3	6	2	6	12	6
$C4$	1	4	8	2	8	16	8
$C5$	1	5	10	2	10	20	10
$C6$	1	6	12	2	12	24	12

Table 2. *Model2* examples

Since the VL path configurations considered cover all possible cases in an industrial network, the conclusion is that VLs in class *II* do not have to be taken into consideration for the computation of vx end-to-end delay distribution.

The resulting reduced simulation space makes it possible to determine an experimental probabilistic upper bound for every VL path in an industrial network. The simulation process considers a specific model for each VL path. Since an industrial network configuration includes more than 6000 paths, this leads to a heavy simulation process. The next section presents a means of speeding up this process. The idea is to model a simplified network architecture which leads to the same end-to-end delay distributions as the original simplified architecture.

3.4 Speeding up the simulation process

This section analyzes the differences between *Model1* and *Model3*, as well as *Model2* and *Model4* in figure 9. More precisely, do certain models with switches $s1_i$ or $s2_j$ connected to $s1$ and $s2$ input ports (typically *Model3* and *Model4*) lead to different vx end-to-end delay distributions than models where each of these switches is replaced by an end system (*Model1* and *Model2*)? If it appears that there is no difference, then *Model3* and *Model4* VL paths can be simplified respectively to the *Model1* and *Model2* ones.

This study follows the same procedure as in the previous section concerning the effective influence of VLs in class *II*. Results are presented in [25] and are summarized in this section.

Let's return to the *Model1* example $P1$ in figure 11. Figure 14 depicts a possible *Model3* configuration corresponding to $P1$, (VLs in class *II* are eliminated, since they have no influence on vx end-to-end delay distribution). End system $e2$ has been replaced by switch $s3$,

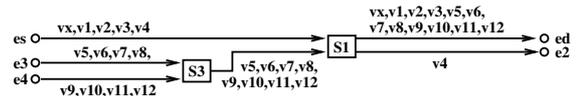


Figure 14. *Model3* configuration

which has two input end systems $e3$ and $e4$. Each of these two end systems emits half of the VLs in class *DI* that were emitted by $e2$ ($v5 \dots v8$ for $e3$ and $v9 \dots v12$ for $e4$).

The vx end-to-end delay distribution is computed for the *Model3* configuration in figure 14. The result is iden-

tical to the one obtained with $P1$. Consequently, in this particular example, the $Model3$ configuration in figure 14 can be simplified to the $Model1$ configuration $P1$.

The complete study is done by considering all $Model1$ and $Model2$ configurations which have been defined for class II influence evaluation (more than two hundred). For each configuration, the following corresponding $Model3$ or $Model4$ configurations are considered:

- each $s1$ or $s2$ input end system ei is replaced by a switch with one input end system emitting all VLs in class DI that were emitted by ei ,
- each $s1$ or $s2$ input end system ei is replaced by a switch with two input end systems, each emitting half of VLs in class DI that were emitted by ei (the case depicted in figure 14,
- each $s1$ or $s2$ input end system ei is replaced by a switch with one input end system per VL in class DI emitted by ei .

The vx end-to-end delay distribution is computed for each $Model1$ or $Model2$ configuration and its corresponding $Model3$ or $Model4$ ones. There is never any significant difference between the distributions computed for a $Model1$ or $Model2$ configuration and the corresponding $Model3$ or $Model4$ ones.

Since the VL path configurations considered cover all possible cases in an industrial network, the conclusion is that the vx end-to-end delay distribution can be computed using its corresponding $Model1$ or $Model2$ configuration.

3.5 Synthesis of the simulation approach

The simulation approach allows a better understanding of avionic flow behavior. In fact, modeling of VLs is an important task based on the taxonomy of VLs defined in the context of existing avionic applications mapped over an industrial AFDX network configuration.

The simplified flow model allows the evaluation of end-to-end delays by queueing network simulation mechanisms. It has been shown that the proposed model simplifications have no observable influence on the obtained end-to-end delay distributions measured on a large set of realistic examples (no counterexample has been found).

The obtained end-to-end delay distributions give important information for the designer about the real behavior of the applications sharing the AFDX network configuration. Moreover, it provides both an experimental upper bound as well as an estimation of the probability to exceed a given bound.

These experimental upper bounds obtained by simulation are not safe, because simulation mechanisms are unable to efficiently take into account rare events. But safe upper bounds are needed for certification purposes. The next section presents the stochastic network calculus approach. The objective of this analytical approach is to provide safe probabilistic upper bounds. It reuses the model-

ing assumptions of the simulation approach and adds pessimistic assumptions on the concurrence of asynchronous flows.

4 The stochastic network calculus approach

This section presents the delay analysis of the AFDX network using a stochastic network calculus approach. The deterministic network theory allows the computation of delay and backlog upper bounds, which have been used for the certification of the AFDX network [10]. Unfortunately, these upper bounds can be very pessimistic as it has been shown that the obtained upper bound can be reached only in the case of a single node architecture [13, 14].

The aim of probabilistic network calculus is to obtain the statistical calculation of delay and backlog bounds. But the computation of a probabilistic upper bound needs extensions of deterministic network calculation concepts and remains a difficult problem. Many studies used probabilistic single node bounds on delay to derive multinode performance bounds by adding the per node bounds. The problem is often the rapid degradation of obtained results as the number of traversed nodes increases. The challenging (and still relatively open) problem is to be able to construct a probabilistic network service curve for a multinode architecture [16]. Few models are however known that allow concatenating probabilistic service curves to derive end-to-end probabilistic network models [12].

The problem addressed in this paper mainly deals with the probabilistic phasing between VLs which is unknown (avionic functions are asynchronous) and conserves the deterministic arrival and service curves defined in the AFDX context. The main problem is to efficiently utilize statistical multiplexing while preserving node concatenation properties [9, 18, 27, 28].

As explained at the beginning of section 3, the problem resides in the evaluation, for a VL path px , of the variable part $WD_{F_{px}}$ of its end-to-end delay (i.e. the waiting times at buffers). For each VL path px , the approach considers its corresponding $Model1$ or $Model2$ configuration, since it has been shown that this simplification has no effect on px end-to-end delay distribution.

The approach is based on results from *Vojnović and Le Boudec* [27, 28]. It proceeds in two parts:

1. the configuration consisting of a VL path px crossing one or several switches and competing with other VLs is transformed into a configuration with a single VL path px crossing a single switch [24],
2. the stochastic network calculus is then applied to this last configuration in order to compute the probabilistic upper bounds [23].

Section 4.1 briefly notes results from *Vojnović and Le Boudec* which allow the computation of an end-to-end delays' probabilistic upper bound in the case of a single flow crossing a single switch (step 2). Section 4.2 details the

transformation process of step 1. Section 4.3 illustrates the approach with some examples of VL paths.

4.1 Probabilistic upper bound in the monoswitch case

This section gives a brief overview of *Vojnović's* and *Le Boudec's* work concerning stochastic analysis on end-to-end delays of flows crossing a single switch [27, 28].

The following notations are used in the remaining of the paper:

- β_X denotes the service curve offered by the switch X to the aggregated flow,
- β_X^f denotes the service curve offered by the switch X to the flow f .
- \tilde{Q} denotes the possible amount of backlog present in the queue.
- Q is the backlog encountered by a given packet at its arrival time,
- $Q(0)$ is the backlog encountered by a packet that arrives at time 0,
- $d(0)$ is the delay incurred by a packet that arrives at time 0.

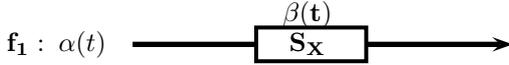


Figure 15. A single flow crossing a switch

Consider the flow f_1 depicted in Figure 15; *Vojnović* and *Le Boudec* have studied and established the lowest stochastic bounds on the output buffer backlog and on the delay to cross switch S_X . Their results can be applied iff the two following assumptions are verified:

- the switch offers to the flow a service curve, denoted $\beta(t) = Rt$ (cf. Figure 15),
- the flow f_1 is regulated at the network ingress point, by a wide-sense increasing function, denoted $\alpha(t)$ (cf. Figure 15).

These two assumptions are true for the AFDX context. Concerning the first assumption, in the AFDX network, each switch can be modeled as a rate-latency function. A rate-latency curve is an affine function, $\beta(t) = R[t - T]^+$, where R is the minimum service offered to input flows, T is the worst-case latency and $[v]^+ = \max(v, 0)$. T might be different from zero in the AFDX context. Therefore, in order to verify the first assumption, T is removed from β before computation and added to the resulting probabilistic upper bound. This assertion is valid since T is a waiting time. Therefore, considering this time before or after the computations leads to the same upper bound.

Concerning the second assumption, each flow f is regulated at its network access by a leaky-bucket function,

$\alpha(t) = \rho t + \sigma$ defined in the following way. σ is the maximum length of a frame generated by the VL, denoted s_{max} . ρ is the VL maximum rate, $\frac{s_{max}}{BAG}$, where BAG is the minimum delay between the emission of two consecutive frames of the VL by its source end system.

Therefore, *Vojnović's* and *Le Boudec's* work can be used to compute the stochastic upper bound on the end-to-end delay of a single flow crossing a single switch in the AFDX context.

In order to determine the f_1 end-to-end delay, the first step is the computation of the backlog at the output buffer crossed by f_1 . *Vojnović* and *Le Boudec* established two results.

The first result concerns the upper bound on the probability that the backlog at the output buffer can exceed a given value. The lowest bound is presented in [27] and given in Theorem 1.

Theorem 1 *If $\rho < R$, for any t , the upper bound of the probability (denoted \mathbb{P}) that the backlog is above a given level b is*

$$\mathbb{P}(\tilde{Q}(t) > b) \leq \sum_{k=0}^{K-1} \exp(-\mathcal{I}g(s_k, s_{k+1})) \quad (1)$$

for any $K \in \mathbb{N}$, and any $0 = s_0 \leq s_1 \leq \dots \leq s_K = \tau$. where τ is the intersection between the arrival curve α and the service curve β : $\tau = \inf\{u \geq 0 \mid \alpha(u) \leq \beta(u)\}$. and where

for $b > \alpha(v) - \beta(u)$, $g(u, v) = +\infty$

for $b < \rho v - \beta(u)$, $g(u, v) = 0$

else $g(u, v) = \frac{\beta(u)+b}{\alpha(v)} \ln \frac{\beta(u)+b}{\rho v} + (1 - \frac{\beta(u)+b}{\alpha(v)}) \ln \frac{\alpha(v) - \beta(u) - b}{\alpha(v) - \rho v}$

The second result concerns the lowest upper bound on the probability that the backlog in the output buffer exceeds a given value at the arrival time of a f_1 frame. This probability is denoted \mathbb{P}_A and named the Palm probability [8]. *Vojnović* and *Le Boudec* have also proved (cf. [27]) Corollary 1.

Corollary 1 *If a packet arrives in the node at time 0, it holds that,*

$$\mathbb{P}_A(Q(0) > b) \leq \frac{R}{\rho} \mathbb{P}(\tilde{Q}(0) > b) \quad (2)$$

An upper bound of the probability $\mathbb{P}(d > u)$ that the end-to-end delay d of the flow f_1 exceeds a given value u is deduced from the Palm probability. The lowest upper bound is established in [27] and recalled in Theorem 2.

Theorem 2 *If a node arrives in node at time 0,*

$$\begin{aligned} \mathbb{P}(d(0) > u) &\leq \mathbb{P}_A(Q(0) > Ru) \\ &\leq \frac{R}{\rho} \mathbb{P}(\tilde{Q}(0) > Ru) \end{aligned} \quad (3)$$

The computation of the stochastic upper bound starts with $u = 0$. Then u is increased until the probability obtained with the previous result is less than a chosen value, for example 10^{-6} .

It then becomes possible to establish the stochastic upper bound for a single AFDX flow crossing a single switch. Obviously, such a configuration is very unusual in an AFDX network. The majority of AFDX flows cross several switches and compete with several other flows. The next section shows how every AFDX flow configuration can be transformed into a single flow crossing a single switch.

4.2 Transformation of multi-hop flows

The transformation process is described considering flow f_1 depicted in Figure 17.a.

The process is based on a result that has been proved for the deterministic network calculus in [19] and summarized by Corollary 2. This result can be applied in the context of AFDX, since all the flows are independent at their ingress point and since the queueing service discipline in the output ports is FIFO.

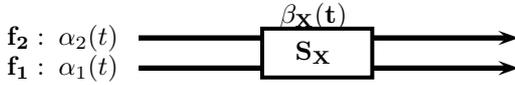


Figure 16. Two flows crossing a switch

Corollary 2 Consider the switch S_X that serves two flows f_1 and f_2 (the flow f_i ($i = 1, 2$)). Each flow f_i has an arrival curve $\alpha_i(t) = \rho_i t + \sigma_i$ (cf. the network configuration depicted in Figure 16)). If the service curve of the aggregated flow is $\beta_X(t) = R[t - T]^+$, then flow f_1 has a service curve

$$\beta_{f_1, S_X}(t) = (R - \rho_2)[t - (T + \frac{\sigma_2}{R})]^+$$

the output flow f_1 , denoted α_1^* , has the curve

$$\alpha_1^*(t) = \rho_1 t + \sigma_1 + \rho_1(T + \frac{\sigma_2}{R})$$

Using Corollary 2, given the arrival curve of all the input flows and the service curve offered by the switch to the aggregate flow, the actual service curve offered to each flow and its output curve can be determined. This result is recursively applied to each crossed switch, until the exact service curves offered to the flow f_1 are obtained.

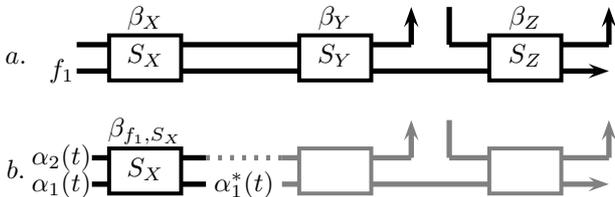


Figure 17. Multi-hops flow transformation

The second step uses another result of [19] concerning the concatenation of switches. If the actual service curve offered to f_1 is known (cf. Figure 18.a), the crossed switches can be concatenated to a single switch (cf. Figure 18.b). The service curve of this new switch is the convolution between the actual service curves.

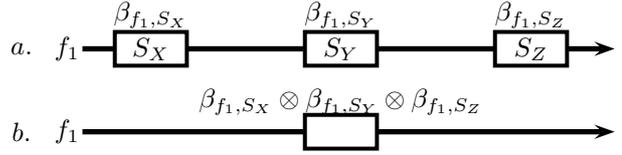


Figure 18. Multi-hops flow transformation

4.3 Example of an evaluation

In this section, the stochastic network calculus approach is applied to the network configurations evaluated by the simulation approach in the process described in section 3.3. Indeed, an important challenge is to estimate the pessimism of the stochastic network calculus approach. The comparison between the results of the two approaches (simulation and stochastic network calculus) measures this pessimism, since the simulation approach closely approximates real network behavior.

Figure 19 depicts the probabilistic upper bounds computed with the stochastic network calculus approach on the six *Model2* configurations of table 2. As with the

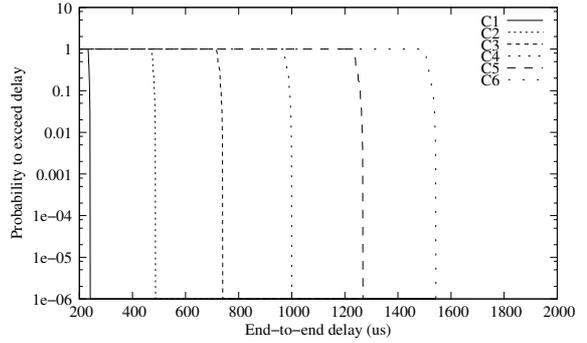


Figure 19. *Model2* analytical probabilistic upper bound

simulation approach, the probabilistic upper bound corresponding to a given probability p increases with the network configuration load. For instance, if $p = 10^{-6}$, the upper bound is between $241 \mu s$ for *C1* and $1543 \mu s$ for *C6*.

The pessimism of the stochastic network calculus approach increases with the network configuration load. If $p = 10^{-6}$, the upper bound for *C1* (lightly loaded configuration, at most 5 % per link) is $237 \mu s$ with the simulation approach compared to $241 \mu s$ with the stochastic network calculus approach. It is $310 \mu s$ compared to $740 \mu s$ for *C3* (a slightly more loaded configuration, at most 15 % per link) and $399 \mu s$ compared to $1543 \mu s$ for *C6* (more heavily loaded configuration, at most 30 % per link).

The set of configurations studied (*Model1*, ..., *Model4*) confirms these differences between the two upper bounds. It is minimal for lightly loaded configurations (under 5 % per link), at most four times the value of the simulation upper bound for 15 % loaded configurations and five times this upper bound for 30 % loaded

BAG (ms)	Number of VL	Frame length (bytes)	Number of VL
2	20	0-150	561
4	40	151-300	202
8	78	301-600	114
16	142	601-900	57
32	229	901-1200	12
64	220	1201-1500	35
128	255	> 1500	3

Table 3. BAGs and frame lengths

Nb of crossed switches	Number of paths
1	1797
2	2787
3	1537
4	291

Table 4. VL paths lengths

configurations. It was mentioned in section 2 that nearly all the links in an industrial network have a load of under 15 %. Thus, the probabilistic upper bound computed by the stochastic network calculus approach is at most about four times the actual upper bound in an industrial network.

5 End-to-end delay analysis examples on an industrial AFDX network configuration

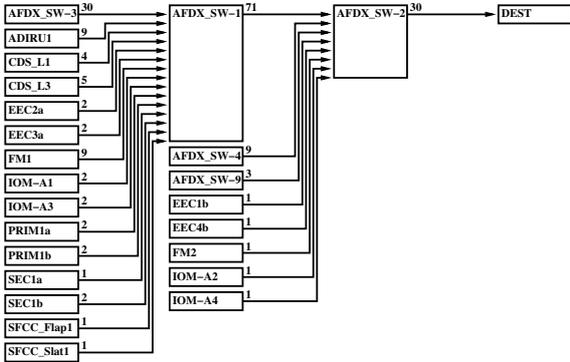


Figure 20. The extracted Virtual Link vl_4

The two approaches presented in this paper have been validated on an industrial AFDX network [10]. It is composed of two redundant networks. Each network includes 123 end systems, 8 switches, 964 Virtual Links and 6412 VL paths (due to VL multicast characteristics). The left part of table 3 gives the dispatching of VLs among BAGs. It can be seen that BAGs are harmonic between 2 and 128. The right part of table 3 gives the dispatching of VLs among frame lengths, considering the maximum length s_{max} . The majority of VLs consider short frames. Table 4 shows the number of VL paths per length (i.e. the number of crossed switches).

The evaluation was conducted on a representative subset of the 6412 paths of the configuration. As an example, figure 20 depicts the model associated with the uni-

Analyzed VL	Path length	Number of DI VLs	Simu UB	SNC UB
vl_1	1	14	166	401
vl_2	1	51	248	560
vl_3	1	143	351	1336
vl_4	2	90	396	1062
vl_5	3	141	603	2047
vl_6	4	228	759	2943

Table 5. Upper bounds for example VLs

cast VL vl_4 . the vl_3 path is $ADIRU1 - AFDX_SW-1 - AFDX_SW-2 - DEST$. Its BAG is 32 ms and its frame length is 343 bytes ($s_{min} = s_{max}$). vl_4 is directly influenced by 90 other VLs emitted by 22 end systems. Each end system executes one or several avionic functions. For instance, $PRIM1a$ is concerned with flight control. A description of all the end systems in figure 20 is beyond the scope of this paper. The upper link load is 14 % (on link $AFDX_SW-1 - AFDX_SW-2$). Thus it is a quite heavily loaded configuration as far as an industrial network is concerned.

Table 5 gives the obtained upper bounds for vl_4 and five other VL paths with various path lengths (between 1 and 4) and different number of directly influencing VLs (from 14 to 228). Considering vl_4 and a probability $p = 10^{-6}$, the experimental upper bound (Simu UB) is 396 μs , compared to 1062 μs for the analytical upper bound (SNC UB). These values confirm the conclusions of section 4.3 (the latter is less than four times larger than the former). The same observation is made on the five other VLs in Table 5. This result has been confirmed for all the studied paths of the industrial configuration.

6 Conclusion

The study presented in this paper concerns the probabilistic analysis of end-to-end delays of an AFDX network. The goal is to compute a probabilistic upper bound for each application flow. Such an upper bound can be exceeded with a given probability p . It is relevant in the context of avionics, since avionic functions are designed to give accurate results even if they miss some frames. Thus, a frame of a given flow may occasionally miss its deadline without any serious consequences on an avionic system. This paper shows how to compute a probabilistic upper bound in the context of industrial AFDX applications.

The simulation approach determines an experimental upper bound. It considers a model of the network configuration and calculates the end-to-end delays of a given flow out of a subset of all possible scenarios. The end-to-end delay distribution of the flow can then be deduced, provided this subset is representative of all possible scenarios. The main challenge of this approach is to extract this representative subset from the huge number of possi-

ble scenarios in an industrial network including approximately one thousand flows. This paper shows how it is possible to focus on the network part relevant for a given flow px . More precisely, it shows that all the flows which never compete directly with px have no influence on its end-to-end delay distribution. Therefore, they do not need to be considered. A simulation model is derived and it provides end-to-end delays distributions, as it closely approximates industrial network behavior. Moreover, it allows the estimation of experimental upper bounds.

The stochastic network calculus approach analytically determines a probabilistic upper bound. It is based on *Vojnović's* and *Le Boudec's* work. For a given flow, it starts from the simplified model defined in the simulation approach context. It has been shown that this model can be transformed into another model consisting of a single flow crossing a single switch. The probabilistic upper bound is then computed on this latter model. This upper bound is a good candidate for certification since it is guaranteed. However, it is often pessimistic, due to the pessimism of network calculus assumptions.

The pessimism of this analytical upper bound can be evaluated on a given network by comparing it with the experimental upper bound. In an industrial AFDX network, the largest difference between the two upper bounds is never more than four times the experimental upper bound obtained by the simulation approach, regardless of the flow considered. It can be much smaller, depending on the load of the links followed by the flow.

An important point is to determine whether it is affordable to have such pessimism in the context of avionics. Clearly, it will be the case if it leads to an acceptable overdimensioning of the network. This has to be evaluated for each new aircraft. An open problem is still the optimization of the probabilistic upper bound obtained by network calculus. First, the degree of probabilism taken into account by our approach has to be precisely analyzed. Second, more recent results on the concatenation of probabilistic arrival and service curves [12, 16] have to be evaluated in the context of an AFDX network. Moreover, other approaches for analytical analysis seem promising, such as the trajectory approach [20] and results on offsets for distributed systems with end-to-end constraints [21]. Open problems are the adequation of these methods and their efficiency in the context of AFDX network.

For future aircraft, the addition of other types of flows (audio, video, best-effort, ...) on the AFDX network is envisioned. These different flows have different timing constraints and criticality levels. Thus, it is necessary to differentiate them and the FIFO policy on switch output ports is not suitable. Thus, it is necessary to consider other service disciplines, such as Static Priority Queueing or Weighted Fair Queueing [22]. Consequently, the two approaches presented in this paper should be extended to cope with these service disciplines.

Moreover, future avionic network architectures will include fieldbuses such as Controller Area Network (CAN)

[17] or FlexRay [7] in addition to AFDX (currently, there are already CAN buses embedded in aircraft). Considering that a flow can be transmitted over more than one technology (e.g. from a CAN station to an AFDX end system), it is necessary to analyze the end-to-end delays over heterogeneous paths. This should include the timing analysis of the bridging strategy between the different technologies.

References

- [1] *ARINC 651, Aeronautical Radio Inc. ARINC specification 651. Design Guidance for Integrated Modular Avionics.*, 1991.
- [2] *ARINC 653, Aeronautical Radio Inc. ARINC specification 653. Avionics application Software Standard Interface.*, 1997.
- [3] *ARINC 429, Aeronautical Radio Inc. ARINC specification 429. Digital Information Transfer System (DITS) parts 1,2,3.*, 2001.
- [4] *ARINC 664, Aircraft Data Network, Part 1: Systems Concepts and Overview.*, 2002.
- [5] *ARINC 664, Aircraft Data Network, Part 2: Ethernet Physical and Data Link Layer Specification.*, 2002.
- [6] *ARINC 664, Aircraft Data Network, Part 7: Deterministic Networks.*, 2003.
- [7] *Flexray Communications System - Protocol Specification - Version 2.0*, 2004.
- [8] F. Baccelli and P. Bremaud. *Elements of queueing theory*. Springer, Berlin, 2000.
- [9] C. Chang, Y. Chiu, and W. Song. On the performance of multiplexing independent regulated inputs. In *ACM SIGMETRICS*, Massachusetts, USA, June 2001.
- [10] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an AFDX network. In *Proceedings of the 18th ECRTS*, Dresde, Germany, July 2006.
- [11] H. Charara, J.-L. Scharbarg, and C. Fraboul. Focusing simulation for end-to-end delays analysis on a switched Ethernet. In *Proceedings of the WiP session of RTSS*, Rio de Janeiro, Brasil, December 2006.
- [12] F. Ciucu, A. Burchard, and J. Liebeherr. A network service curve approach for the stochastic analysis of networks. *ACM SIGMETRICS performance evaluation review*, 33:279–290, June 2005.
- [13] R. Cruz. A calculus for network delay, part I. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.
- [14] R. Cruz. A calculus for network delay, part II. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.
- [15] J. Diaz, J. Lopez, M. Garcia, A. Campos, K. Kim, and L. Lo Bello. Pessimism in the stochastic analysis of real-time systems: concept and applications. In *Proceedings of the 25th IEEE International Real-Time Systems Symposium*, December 2004.
- [16] M. Fidler. An end-to-end probabilistic network calculus with moment generating functions. In *In IEEE 14th International Workshop on Quality of Service (IWQoS)*, pages 261–270, New Haven, June 2006.
- [17] Road vehicles – controller area network (can). ISO Standard 11898, International Organization for Standardization, 1993.

- [18] G. Kesidis and T. Konstantopoulos. Worst-case performance of a buffer with independent shaped arrival processes. *IEEE communication letters*, 4, January 2000.
- [19] J.-Y. Le Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001. ISBN: 3-540-42184-X.
- [20] S. Martin and P. Minet. Schedulability analysis of flows scheduled with fifo: application to the expedited forwarding class. In *IPDPS*, 2006.
- [21] J. Palencia and M. Gonzalez Harbour. Offset-based response time analysis of distributed systems scheduler under edf. In *ECRTS*, 2003.
- [22] A. Parekh and R. Gallager. A generalised processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE transactions on networking*, June 1993.
- [23] F. Ridouard, J. Scharbarg, and C. Fraboul. Stochastic network calculus for end-to-end delays distribution evaluation on an avionics switched ethernet. In *Proceedings of INDIN*. IEEE, July 2007.
- [24] F. Ridouard, J. Scharbarg, and C. Fraboul. Stochastic network calculus for multi-switches flows on an avionics switched ethernet. In *Proc. of the 7th International Conference on Fieldbuses and networks in industrial and embedded systems*. IFAC, November 2007.
- [25] J.-L. Scharbarg and C. Fraboul. A generic simulation model for end-to-end delays evaluation on an avionics switched Ethernet. In *Proc. of the 7th International Conference on Fieldbuses and networks in industrial and embedded systems*, Toulouse, France, November 2007. IFAC.
- [26] J.-L. Scharbarg and C. Fraboul. Simulation for end-to-end delays distribution on a switched ethernet. In *Proc. of the 12th International Conference on Emerging Technologies and Factory Automation*, Patras, Greece, September 2007. IEEE.
- [27] M. Vojnović and J. Le Boudec. Stochastic analysis of some expedited forwarding networks. in *Proceedings of Infocom, New-York*, June 2002.
- [28] M. Vojnović and J. Le Boudec. Bounds for independent regulated inputs multiplexed in a service curve network element. *IEEE Trans. on Communications*, 51(5), May 2003.