

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers

Computer Science and Engineering, Department
of

2011

Scheduling Design and Analysis for End-to-End Heterogeneous Flows in an Avionics Network

Yu Hua

University of Nebraska-Lincoln, yhua@cse.unl.edu

Xue Liu

University of Nebraska-Lincoln, xueliu@cse.unl.edu

Yu Hua

Huazhong Univ. of Sci. & Tech., csyhua@hust.edu.cn

Follow this and additional works at: <https://digitalcommons.unl.edu/cseconfwork>

Hua, Yu; Liu, Xue; and Hua, Yu, "Scheduling Design and Analysis for End-to-End Heterogeneous Flows in an Avionics Network" (2011). *CSE Conference and Workshop Papers*. 222.

<https://digitalcommons.unl.edu/cseconfwork/222>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Scheduling Design and Analysis for End-to-End Heterogeneous Flows in an Avionics Network

Yu Hua^{1,2}Xue Liu¹¹University of Nebraska-Lincoln, Lincoln, NE, USA

{yhua, xueliu}@cse.unl.edu

²Huazhong University of Science and Technology, Wuhan, China

csyhua@hust.edu.cn

Abstract—Avionics Full Duplex (AFDX) Switched Ethernet technology provides a deterministic network with guaranteed service to support real-time data transmission in real-world avionics applications. The determinism provides a worst-case upper bound of end-to-end transmission delays of virtual links (VLs) that are often assumed to be homogeneous and have similar transmission requirements. However, performing the analysis of end-to-end delays of heterogeneous flows remains an open problem. This paper derives end-to-end delay bounds of transmitting heterogeneous flows, including avionics, multimedia (video & audio) and best-effort data flows, in an AFDX network that uses Deficit Round Robin (DRR) scheduling policy on switch output ports. To this end, we transform scaled multi-type flows to a single representation of utilization, i.e., DRR quanta, to efficiently handle heterogeneous flows in a unified way and comprehensively study their end-to-end delays. We further compare the DRR-based scheduling policy with current avionics standards, i.e., FIFO and a static priority policy, in terms of transmission delays and fairness. Extensive experiments based on periodical and sporadic flows in an AFDX prototype show the efficacy and efficiency of our proposed schemes. To the best of our knowledge, this is the first work that analyzes end-to-end delays based on the DRR policy for heterogeneous flows in an AFDX network.

I. INTRODUCTION

Avionics Full Duplex (AFDX) Switched Ethernet has recently become a standard to support real-time data transmission among avionics subsystems. This standard builds upon protocol specifications of IEEE 802.3 [1] and ARINC 664 [2] and eliminates the potential indeterminism of conventional Ethernet transmission to avoid frame losses. The indeterministic problem unfortunately is not completely solved but shifted to switch level where heterogeneous flows compete for available resources at the switch, thus resulting in larger delays, larger jitters and larger unfairness [3]. Hence, in order to improve the overall performance, the main challenge is to efficiently and fairly schedule heterogeneous flows that are transmitted upon redundant links in an AFDX network.

A typical AFDX network generally consists of avionics subsystems, interconnect networks and source/destination end systems. Specifically, the avionics subsystems include traditional on-board aircraft system, such as global position system (GPS) and flight control system (FCS). The interconnect networks use a full-duplex switched Ethernet consisting of links and switches to support data exchange among different

avionics systems. The end systems actually serve as an interface between the subsystems and the interconnect networks to guarantee real-time and reliable data transmission by using deterministic Virtual Links (VLs).

A virtual link constructs a virtual communication connection from one source end system to one or more destination end systems, forming a mono-sender multicast path. According to the AFDX specification [2], we can identify a VL by setting its available 16-bit ID, Bandwidth Allocation Gap (BAG) and the largest length of VL frames (i.e., L_{max}), where the BAG represents the minimum interval between two consecutive frames sent to a VL. An AFDX network specifies the BAG duration from 1ms to 128ms to serve as the bandwidth control mechanism for virtual links.

In order to guarantee transmission reliability in an AFDX network, one of the most important characteristics is the redundant management on virtual links. Specifically, an AFDX network constructs two independent paths between end systems and redundant switches to protect the network from a failure at the MAC level. The same frame hence needs to be transmitted across two independent networks. In order to simplify the operations at the destination end system, a redundant copy of a frame should be sent within a maximum time difference of 0.5ms at the source end system [2]. The destination end system only accepts the first valid frame and discards the redundant one. The parameter in redundancy management is the *SkewMax*, which represents the maximum time between the reception of the original frame and its redundant copy. The value of *SkewMax* depends upon the network topology, i.e., the number of switches crossed by the transmitted frames, and is defined by the system administrator. Since the frames transmitted within an AFDX network come from heterogeneous flows with different priorities and the end-to-end delays meanwhile depend upon the configurations of two independent networks, performing end-to-end delay analysis becomes very challenging. Even the implementation of a simulation platform requires non-trivial work, which is helpful to study the end-to-end delays.

End-to-end delays consist of the sum of transmission delays on links and latencies in switches and end systems, usually depending upon network configurations, such as available bandwidth, pre-defined frame length and scheduling policies. The dominant delays come from the longest waiting service

time within frame queues where FIFO policy is often used, assuming that all VLs have the same priority to obtain the resource *utilization*. The utilization is interpreted as the resource consumption rate within the given time interval. Each VL flow can transmit frames when it obtains enough utilization. However, it is difficult to accurately obtain quantitative representation of multi-type flows due to their heterogeneity. Existing FIFO policy equally treats each flow, which makes it inefficient for the AFDX network due to ignoring the differentiated requirements from heterogeneous flows. Furthermore, the AFDX network in the envisioned future involves multi-type flows, such as avionics, multimedia (video & audio) and best-effort data, to monitor real-time working status and manipulate avionics. For instance, a pilot needs to check navigation videos from electronic cameras, monitor real-time data from installed sensors and discuss with other flight crew through audio voice, thus requiring reliable and real-time services for heterogeneous flows that have differentiated requirements of transmission delays. Therefore, we need to study the end-to-end transmission delays of fairly scheduling heterogeneous flows based on quantitative utilization.

The analysis of end-to-end delays tightly relies on the scheduling algorithms. The main function of a scheduling algorithm is to select the next transmitted packet and transmission time while taking into account the required performance metrics [4], [5]. Existing work-conserving scheduling algorithms can be classified into two categories, i.e., sorted-priority and frame-based. The former first allocates a timestamp to each queued packet and then transmits the packets in the order of increasing timestamp. The latter divides time into frames and a packet to transmit is selected in a per-frame basis. Round-robin algorithms belonging to the frame-based class cyclically provide the service for various flows.

In order to guarantee the bounds of end-to-end delays together with fairness for heterogeneous flows, we need to carefully design an efficient scheduling algorithm. This paper selects Deficit Round Robin (DRR) [6], rather than conventional FIFO and static priority, to study the end-to-end delays in an AFDX network due to two reasons. First, the DRR scheduling has been widely used in many real-world applications. DRR [7], [8] is one of the most popular frame-based round robin schedulers and was shown to be able to reduce per-packet computation cost, with worst-case per packet $O(1)$ complexity. This indicates that the number of operations required for selecting the next transmitted packet is constant with respect to the number of flows, thus obtaining significant time saving and fairness. Second, the DRR policy makes use of a unified and quantitative approach to scheduling data flows. Their differentiated transmission requirements are represented as the single number of allocated quanta. We leverage this property in our analysis of the AFDX network. We transform the problem of scheduling heterogeneous flows in an AFDX network into a relatively simpler problem of quanta computation by using the DRR scheduling.

In this paper, we study the end-to-end delays for scheduling heterogeneous flows by using DRR scheduler with the aid of

network calculus [9]. Note that the delay bounds of an AFDX network using DRR scheduler are highly versatile and can be tailored for delay analysis of other cyber-physical networks. This paper makes the following contributions.

- **End-to-end delays of heterogeneous and redundant flows.** Current standards in an AFDX network [2] mainly take into account the FIFO and static priority scheduling policy. We make use of DRR scheduler to help study the end-to-end delays of heterogeneous flows, while taking into account redundant management of transmitting the same frame within two independent paths. The basic idea is to unify the utilization requirements for transmitting heterogeneous flows into quanta of a DRR scheduler.
- **Prototype Implementation.** In order to examine the performance of our proposed scheduling scheme, we build a prototype of AFDX network to conduct extensive simulations and evaluate the end-to-end delays for heterogeneous flows by using FIFO, static priority (e.g., RMA, Rate Monotonic Analysis [10]) and DRR schedulers. The components and interfaces of the prototype resemble those in typical AFDX networks. Our simulation further takes into account the periodic and sporadic flows and the experimental results verify the efficacy and efficiency of the proposed schemes.

The rest of the paper is organized as follows. Section II introduces the AFDX network with the DRR-based scheduling policy. Section III discusses the analysis of DRR-based end-to-end delays. Section IV shows the performance evaluation. Section V shows the related work. Finally, Section VI concludes our paper.

II. AFDX NETWORK BASED ON DRR SCHEDULING

An AFDX network provides real-time and reliable transmission service. It can obtain deterministic end-to-end delays due to its static definition of virtual links (VL) routing paths. A typical configuration of AFDX network as shown in Figure 1 consists of end systems, multiple interconnected switches and VLs. Each VL is characterized by its Bandwidth Allocation Gap (BAG) and L_{max} parameter (i.e., the largest length of VL frames). The end systems serve for the input and output of entire network. A switch has no buffer at its input ports and one FIFO buffer at each output port that ignores differentiated transmission requirements from heterogeneous flows. The FIFO buffer thus becomes potential transmission bottleneck due to queuing delays, resulting in inefficiency and unfairness for scheduling heterogeneous flows that are aggregated into VLs. In this paper, heterogeneous flows refer to the transmitted multi-type flows that have different requirements for bandwidth and delays.

Heterogeneous flows are prevalent in an AFDX network. Typical heterogeneous flows include avionics, multimedia and best-effort data. AFDX network aggregates data flows from output ports into VLs to carry out deterministic routing. A typical AFDX network forms a mono-sender multicast path: the VL originating from a single end system transmits data to a fixed set of end systems, actually producing a multicast

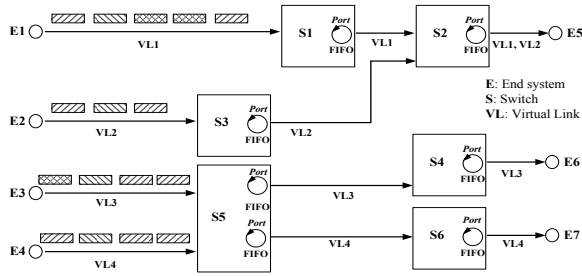


Fig. 1. A typical AFDX network with FIFO scheduler.

path. The competition among multiple VLs at output ports possibly introduces traffic congestion and extra end-to-end delays due to information loss and re-transmission. Therefore, this paper proposes DRR-based design for scheduling multiple heterogeneous flows to obtain fairness and guaranteed end-to-end delays.

A. Scheduling Design

We propose DRR-based scheduling policy in an AFDX network to efficiently support fair scheduling for multiple heterogeneous flows with bounded end-to-end delays. Figure 2 shows an example of the AFDX network using DRR schedulers, rather than FIFO schedulers, at output ports in switches. Each end system sends multi-type flows, including avionics, multimedia (video & audio) and best-effort data, which are further aggregated into VLs. VLs then transmit these multi-type flows that have different transmission requirements, which are represented as different amounts of quanta.

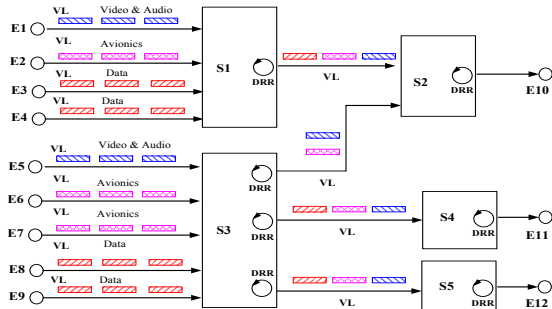


Fig. 2. AFDX network with DRR scheduler for heterogeneous flows.

The VLs follow standard DRR operations at switch as defined in ARINC 664 [2] and however, the main difficulty to efficiently transmit multi-type flows comes from their heterogeneity that introduces relatively high operation complexity for unified scheduling, i.e., by using a single utilization to carry out quantitative evaluation for end-to-end delays. Recall that the utilization is the resource consumption rate within the given time interval. We handle this problem by using DRR scheduler for incoming flows at output ports to guarantee fairness among all transmitted flows and decrease potential traffic congestion.

B. Unified Utilization in DRR Scheduling

Deficit Round-Robin (DRR) [6] is a variation of Weighted Round-Robin (WRR) [11] in order to allow flows with variable packet lengths to fairly share available bandwidth, i.e., obtaining the utilization corresponding to their requirements.

The DRR characterizes each flow by a quantum allocated in advance and by a deficit variable. The quantum viewed as utilization shows the quantity of packets that a flow should ideally transmit during a round. The deficit variable measures the available quantum at the current round, i.e., the balance of utilization. Specially, in each round, DRR allows a flow to transmit packets no more than the sum of the allocated quantum and deficit variable. After servicing for a backlogged flow, DRR decreases its deficit by the number of transmitted bits. When a flow can not transmit a packet due to too large size in the current round, its allocated quantum will be added to the flow's deficit variable for the next round [12].

We use DRR policy to represent utilization of heterogeneous flows in a unified way to facilitate the analysis of end-to-end delays. Utilization is generally defined as the resource consumption rate within measured time interval [13], which is also called as task periods. However, since not all tasks are periodic in real-world applications, existing work mainly makes use of relative deadlines to represent the measured intervals and has to introduce extra restrictive constraints, being unable to provide versatile utilization bounds for scheduling models. In order to handle this issue and further provide the fair scheduling for heterogeneous flows, we present a unified utilization policy based on DRR scheduler, in which we formulate utilization of different flows into their allocated quanta. The quanta in fact indicate the amounts of resources to be used by a flow in each round. Since the utilization of all heterogeneous flows can be easily represented as the single quantum, we can schedule these VL flows in a unified way by adjusting their associated quanta. The transmission requirements from heterogeneous flows are then transformed into the amounts of allocated quanta. For example, a multimedia flow exhibiting higher priority than best-effort data can be represented as the description that the former obtains larger quanta than the latter. We further formulate the DRR-based scheduling and study the end-to-end delays with the aid of network calculus.

III. END-TO-END DELAY ANALYSIS

In this section, we study the end-to-end delays of transmitting heterogeneous flows through multiple intermediate switches where DRR scheduling is used at output ports.

Network calculus [9] is a min-plus based system theory where addition becomes computation of minimum and multiplication becomes addition. Network calculus in essence makes use of convolution formula to analyze the end-to-end performance of a single server and concatenation of tandem servers due to their “pay-bursts-only-once” property that builds upon a multi-node scenario to produce the end-to-end bounds. The property comes from the observation that a burst may occur in any node of an end-to-end path, but absolutely not in all nodes at the same time, thus particularly facilitating the analysis on a long end-to-end path.

Considering a system S , a cumulative function $R(t)$ represents input data on the flow in the time interval $[0, t]$ with initialization $R(0) = 0$. $R^*(t)$ is the corresponding output

function of system S . Network calculus uses real-valued, non-negative, and wide-sense increasing function \mathcal{F} .

$$\mathcal{F} = \{\forall t \geq s \geq 0, f: \mathbb{R}^+ \rightarrow \mathbb{R}^+, f(t) \geq f(s), f(0) = 0\}.$$

In our analysis, the end-to-end delays are first decomposed into several parts, including the delays from source end systems, transmission links, switches and destination end systems, which are then studied respectively. Our analysis leverages upon network calculus. Specifically, we make use of leaky bucket function as arrival curve, i.e., $\alpha(t) = rt + b$ where b is the burst and r is the rate, to generate (b, r) -constrained flows. The service curve is rate latency function, i.e., $\beta(t) = R(t - T)$ with latency T and rate R .

A. Redundant Frame Transmission

An AFDX network improves upon system reliability by transmitting each frame into two independent switched networks. The destination end system thus receives two copies of each frame. The redundancy management mechanism within AFDX helps identify redundant copy by checking the pre-defined sequence number on a virtual link.

In order to decrease the potential interference among multiple virtual links using the same physical link, an AFDX network limits the transmission rate of frames on a virtual link, represented as BAG, to explicitly regulate the minimum transmission interval between successive frames. On the other hand, since the original frame and its redundant copy are sent within maximum time interval of 0.5ms, they hence produce an upper bound of 0.5ms delay at the source end system. Note that the BAG interval is used to shape multiple frames and the 0.5ms time interval is used for the original frame and its redundant copy.

B. End-to-End Delay Composition

The end-to-end delay of a VL, called $E(D_{VL})$, can be characterized by the devices that the VL passes through. Hence the total delay is the sum of all the delays of individual devices including source end system, transmission links, switches and destination end system, i.e., $E(D_{VL}) = D_{source} + D_{transmit} + D_{switch} + D_{destination}$.

- D_{source} is the processing delay in source end system while generating (b, r) -constrained flows as shown in Section III-C. The source end system needs to first select a transmitted frame from a virtual link queue, then assign the per-VL sequence number, further replicate the frame in the redundant management and finally transmit the frame and its copy on the physical links.
- $D_{transmit}$ is the transmission delay over the links and mainly consists of the number of links n_l , available bandwidth b_{n_l} and frame size F_{VL} , i.e., $D_{transmit} = \sum_{i=1}^{n_l} (b_{n_i}) * F_{VL}$, which in practice depends upon environment settings and becomes one constant within specific network configuration.
- D_{switch} is the delay within the switches from source to destination end systems. Specifically, the delay can be further divided into two parts, technological delay, d_t ,

which is bounded by a *constant* for specified hardware and switch capacity [14], and queuing delay, D_{DRR} , which depends upon scheduling strategy. We make comprehensive analysis of delays using DRR policy as shown in Section III-D.

- $D_{destination}$ is the delay in the destination end system where each arriving frame needs to pass through integrity checking and redundancy management, since source end system generally sends each frame twice, i.e., frame F_1 and F_2 , with the same sequence number to transmit within two independent networks to obtain high reliability. The end-to-end delays are respectively represented as D_{VL1} and D_{VL2} . In addition, the destination end system allows to wait at most *SkewMax* time interval [2] for receiving redundant frame.

According to the above composition, we further study each part of the end-to-end delays for transmitting heterogeneous flows by using DRR scheduler in an AFDX network.

C. Delay from Source End System

The delay in source end system depends upon arrival curves that produce (b, r) -constrained flows with different priorities.

Theorem 1. (Source End System Delay) Consider n arrival non-preemptive flows with arrival curves $\alpha_1, \alpha_2, \dots, \alpha_n$ that are (b_i, r_i) -constrained ($i = 1, 2, \dots, n$) with static priorities. By using redundant management in an AFDX network, the source end delay D_{source}^i of flow i is:

$$D_{source}^i = \frac{1}{2} \left(\frac{\sum_{j: \omega_j \Delta \geq \omega_i \Delta} b_j + L_{i, \max}}{C - \sum_{j: \omega_j \Delta > \omega_i \Delta} r_j} + 0.5ms \right) \quad (1)$$

where 0.5ms is the maximum time interval of original frame and its copy at the source end system, $L_{i, \max}$ is the maximum length of packets in flow i , C is the output capacity, Δ is the standard allocated quantum and ω_i is scaling factor.

Proof: When a flow i with static priorities arrives, we identify the priority by checking its allocated quantum $\omega_i \Delta$. The processed packets contain the bursts from higher-priority flows, i.e., $\sum_{j: \omega_j \Delta \geq \omega_i \Delta} b_j$, and its own maximum packet, i.e., $L_{i, \max}$. On the other hand, the allocated bandwidth for flow i is $C - \sum_{j: \omega_j \Delta > \omega_i \Delta} r_j$. Thus, for a single flow i , its delay can be bounded by $\frac{\sum_{j: \omega_j \Delta \geq \omega_i \Delta} b_j + L_{i, \max}}{C - \sum_{j: \omega_j \Delta > \omega_i \Delta} r_j}$.

AFDX network generally makes use of redundant management to transmit packets for high reliability and specifically: two VL frames need to be sent with the maximum time interval, i.e., 0.5ms. Thus, we can obtain the delay of flow i with redundant management is $(D_{source} + 0.5ms)/2 = (\sum_{j: \omega_j \Delta \geq \omega_i \Delta} b_j + L_{i, \max})/2(C - \sum_{j: \omega_j \Delta > \omega_i \Delta} r_j) + 0.5ms/2$. ■

D. DRR-based Switching Delay

AFDX network utilizes DRR scheduler to support the transmission of heterogeneous flows essentially by allowing the remaining quantum from previous rounds to be used for the next round. DRR exhibits $O(1)$ complexity if the allocated

quantum of each flow is no smaller than its maximum packet size. Specifically, in this paper, the allocated quanta are scaled by the tunable factor ω according to their heterogeneous types, rather than active list number [7] and relative deadline [13].

The DRR uses round-robin fashion to provide fair transmission service and guaranteed bounds of end-to-end delays. Initially, the deficit counter θ_i of flow i is set to 0 and the counter θ_i is increased by a quantum Δ that is further scaled by ω_i . The selected flow i thus obtains $\omega_i\Delta$ quantum for data transmission. The flow i totally has $\theta_i + \omega_i\Delta$ allowable transmitted length each round. When the current packet size in the flow i is larger than $\theta_i + \omega_i\Delta$, the packet can not be sent in this round. However the allocated quantum $\omega_i\Delta$ is saved and added into deficit counter for next round. When a packet is transmitted, the deficit counter is decreased by the packet length. DRR can also be considered as a latency-rate server [15], characterized by its worst-case delay and guaranteed service rate, in order to provide affine services. Due to space limitation, we ignore other details of DRR that can be found in Ref. [6]–[8].

Theorem 2. (DRR Scheduling Delay) *Given time interval $[t_1, t_2]$, the upper bound of DRR scheduling delay D_{DRR}^i for flow i is*

$$\frac{1}{C} \left[\sum_{j=1, j \neq i}^n \left(\frac{L_i}{\omega_i \Delta} + 1 \right) \omega_j \Delta + \sum_{j=1, j \neq i}^n L_{j, \max} + L_{i, \max} \right] \quad (2)$$

where C is the output capacity, ω_i is scaling factor, Δ is the standard allocated quantum and $L_{i, \max}$ is the maximum length of packets in flow i during time interval $[t_1, t_2]$.

Proof: According to the conclusion in [6], two flows i and j must have obtained one round-robin opportunity for data transmission between two opportunities if they are backlogged during time interval $[t_1, t_2]$, represented as $|\text{opportunity}(i) - \text{opportunity}(j)| \leq 1$. Meanwhile, a packet in flow i with length L_i can obtain scheduling service if the opportunity numbers of all flows are $\lceil \frac{L_i}{\omega_i \Delta} \rceil + 1$. Thus, during interval $[t_1, t_2]$, flow i allows to send at most $\lceil \frac{L_i}{\omega_i \Delta} \rceil \omega_i \Delta + L_{i, \max}$ packet length. Furthermore, a flow j ($j \neq i$) is able to send at most $(\lceil \frac{L_i}{\omega_i \Delta} \rceil + 1) \omega_j \Delta + L_{j, \max}$ packet length. Therefore, when a packet with length L_i is transmitted, the total transmitted packets for n flows during interval $[t_1, t_2]$ are

$$\sum_{j=1, j \neq i}^n \left(\frac{L_i}{\omega_i \Delta} + 1 \right) \omega_j \Delta + \sum_{j=1, j \neq i}^n L_{j, \max} + L_{i, \max}$$

When further taking into account the available output capacity C , we can obtain the final result. ■

The above result we just show generalizes previous results [6] and [7]. When setting $\omega_i = 1$ and $\omega_i \Delta = L_{i, \max}$, we can respectively obtain delay results in [6] and [7], which are the specified cases of our analysis.

We further study the DRR-based service curve in a single switch and then extend it into multi-hop scenarios by considering “pay-bursts-only-once” property of network calculus [9].

Theorem 3. (DRR-based Service Curve) *Consider a switch serving for n arriving flows. Its service curve using DRR-based scheduling in a single node (i.e., 0-hop) during time interval $[t_1, t_2]$ is $\beta_{DRR}^0 = R_{DRR}^0(t - T_{DRR}^0)$, where $R_{DRR}^0 = \sum_{i=1}^n \omega_i \Delta_{t_2 - t_1} / (t_2 - t_1)$ and $T_{DRR}^0 = \sum_{i=1}^n D_{DRR}^i$.*

Proof: A standard service curve is $\beta = R(t - T)$ where R and T are respectively service rate and delay. During time interval $[t_1, t_2]$, the ideal number of transmitted frames through DRR is $\sum_{i=1}^n \omega_i \Delta_{t_2 - t_1}$, thus obtaining the R_{DRR}^0 . Furthermore, the T_{DRR}^0 is the sum of all flows through DRR-based scheduler based on Equation 2. ■

We consider the multi-hop transformation where flows transmit through multiple switches by using “pay-bursts-only-once” property of network calculus [9]. We study the end-to-end delays across multiple switches as shown in Figure 3. The concatenated switches function as a single one by taking into account the convolution operations since a burst can not occur in all switches at the same time.

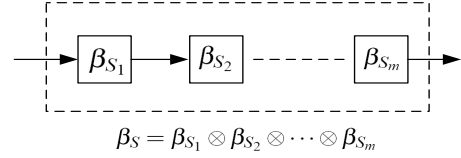


Fig. 3. A “pay-bursts-only-once” scenario.

Theorem 4. (Pay-Bursts-Only-Once Concatenation) *Assume a flow traverses two servers with service curves $\beta_1(t)$ and $\beta_2(t)$ respectively. The concatenation of two servers is equivalent to a single server system with service curve $\beta(t) = (\beta_1 \otimes \beta_2)(t)$.*

The theorem can be easily proved by using the association of min-plus convolution, i.e., $((R \otimes \beta_1) \otimes \beta_2)(t) = (R \otimes (\beta_1 \otimes \beta_2))(t)$. The concatenation theorem in fact exhibits “pay-bursts-only-once” property that allows the end-to-end delays and backlog bounds to be scaled linearly in the number of multiple servers, rather than quadratic scaling in summing up single servers.

The upper bound of the end-to-end delay is tightly correlated with the transmission time and the aggregate waiting time in queues. The network calculus can help analyze the upper bounds of end-to-end delays of elementary entities in the AFDX network.

Corollary 5. (Switches Concatenation) *Consider a VL flow that routes through a switch set $S = \{S_1, S_2, \dots, S_m\}$ represented as service curves β_k , ($k = 1, \dots, m$). The switch set can be concatenated into a single switch by using convolution among all switches $\beta_S = \beta_{S_1} \otimes \beta_{S_2} \otimes \dots \otimes \beta_{S_m}$.*

Corollary 6. (Multi-hop Switching Delay) *Consider a VL flow i routing through m switches. The total delay from intermediate switches, D_{switch} , is*

$$\sum_{k=1}^m \left(T_{DRR}^k + \frac{\max_{\omega_i^k > \omega_j^k} \{L_{j, \max}^k\}}{R_{DRR}^k} + \frac{L_{i, \max}^k}{\min_{j \neq i} (R_{DRR}^k - r_j)} \right) \quad (3)$$

We hence obtain the delays of a flow before it arrives at the destination, called “Before-destination Delay”, to facilitate further analysis in the destination end system.

Definition 1. (Before-destination Delay) Before arriving at destination end system, the delay of each VL flow is $D_{VL} = D_{source} + D_{transmit} + D_{switch}$.

E. Destination End Delay

Destination end system needs to spend processing time, called D_{dest} , on carrying out integrity check to guarantee accurate transmission and redundant management to remove duplicate copies. However, since two frames with the same sequence number transmit within independent networks and route in different paths, the time arriving at destination end system depends upon device configuration and link status, thus inducing some randomness and becoming difficult to accurately predict in advance. Parameter $SkewMax$ therefore defines the maximum allowable time between valid frames according to the specification in an AFDX network [2]. When $SkewMax$ value for a VL is exceeded, the integrity check is reset to accept the next valid frame, regardless of its sequence number.

F. End-to-End Delay

The end-to-end delay of a frame transmitted in a VL, represented as $E(D_{VL})$, essentially depends upon the redundant transmission in an AFDX network. Specially, the end-to-end delay for a frame is the sum of delays experienced at all hops from the source to the destination end systems. The lower bound of delay is approximated to $D(h_1)$, which assumes that a frame in a VL transmits through totally h_1 hops and is processed at intermediate nodes without queueing delays. On the other hand, the upper bound is represented as D_{VL} as shown in Definition 1. In practice, an AFDX network makes use of redundant transmission to guarantee the reliability and protect the network from a failure. Figure 4 shows the redundant end-to-end transmission delays. The $F(VL_1)$ and $F(VL_2)$ respectively represent the random and independent distribution of their end-to-end delays with a maximum $0.5ms$ time difference at the source end system. At the destination end system, the membership of a frame that has first arrived can be maintained for a maximum time, i.e., $SkewMax$, to discard its redundant one that transmits through another path. Figure 4 uses $Arrive(VL_1)$ and $Arrive(VL_2)$ to respectively represent the random arriving time of flow VL_1 and VL_2 .

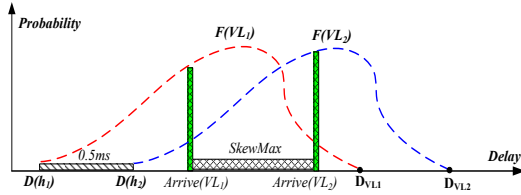


Fig. 4. Redundant end-to-end transmission delays for flows VL_1 and VL_2 .

Therefore, based on the conclusion in Equation 1 and 2 and Definition 1, we obtain the end-to-end delay using DRR scheduling in an AFDX network.

Corollary 7. (End-to-End Delay) The upper bound of end-to-end delay of a flow transmitting through an AFDX network is

$$E(D_{VL}) = \begin{cases} D_{VL_1} + D_{dest} + SkewMax, & \text{if } 0.5ms > SkewMax \\ D_{VL_2} + D_{dest}, & \text{if } 0.5ms \leq SkewMax \end{cases}$$

where D_{VL_1} and D_{VL_2} respectively represent the delays of original and redundant VL flows, $SkewMax$ is maximum time between reception of valid frames.

Proof: The upper bound of end-to-end transmission delays depends upon the parameter setting, i.e., $0.5ms$ and $SkewMax$. When $0.5ms$ is larger than $SkewMax$, the first arriving VL_1 needs to first spend a D_{dest} for processing arriving frame and then wait a $SkewMax$ to complete the end-to-end transmission. On the other hand, when $0.5ms$ is smaller than $SkewMax$, the maximum value of end-to-end delay comes from the VL_2 and the upper bound is naturally D_{VL_2} plus the processing time D_{dest} . ■

G. Parameter Analysis for Quanta

One key issue in an AFDX is to accurately determine the quanta of heterogeneous flows for DRR-based scheduling. The quanta allocation is non-trivial since it heavily depends upon the specified network configuration and unpredictable bursts of network traffic. If the quanta is too large, the flow allowing to use will occupy the scheduler for a long time and other flows have to wait, thus resulting in long delays and potential packet loss. On the other hand, if the quanta is too small, the scheduler will run for many rounds, in which few packets can be transmitted, and most flows have to wait to obtain large enough quanta, also leading to potential delays and packet loss. We make use of an experimental methodology, i.e., the popular sampling-based approach [16], [17], to identify the optimal values for facilitating the analysis of the end-to-end delays. Specifically, we first construct a simulation platform according to the specifications in real-world avionics applications [2], [3] to identify the optimal quanta for multiple heterogeneous flows, represented as different ω values. The sampling-based approach examines end-to-end delays under different ratios of allocated quanta, when taking into account a heavy-load star topology that runs at 100Mbps.

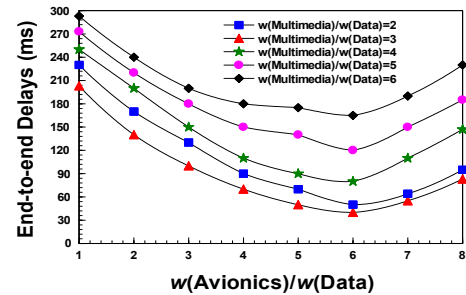


Fig. 5. End-to-end delays for different ratios of allocated ω values.

Figure 5 shows the average end-to-end delays of heterogeneous flows in terms of increased ratios of $\omega(Avionics)$ to $\omega(Data)$ and $\omega(Multimedia)$ to $\omega(Data)$. We select the

optimal ratio of three typical flows to be 6:3:1, respectively for avionics, multimedia and best-effort data to facilitate the following experiments.

IV. PERFORMANCE EVALUATION

In this Section, we present the experimental results of our proposed scheme that uses DRR-based scheduler to efficiently support the scheduling for heterogeneous flows. Current standards of AFDX network only support the FIFO [18] and Static Priority (SP) scheduling policies (e.g., RMA, Rate Monotonic Analysis [10]). The RMA generally exhibits pessimistic worst-case estimate to determine whether a particular set of tasks can be scheduled in a given environment. We compare DRR with them to examine the performance improvements and guide the further implementation in real-world devices.

A. Simulation Setup

Our simulations are configured according to the specifications of the AFDX network [2]. Our evaluation is in terms of end-to-end delays and scheduling fairness. Fairness is an important performance metric for many network applications. In this paper, given n heterogenous flows, the fairness metric is defined in Section IV-B2. It represents the difference of resource utilization rates of flows. It is worth noting that fairness is not a first class concern for real-time applications, while guaranteeing the delay to meet deadline is the most important concern. However, given all flows meet the deadlines, fairness is still an important concern. In addition, the prototype runs on the Linux kernel environment that uses 3.2GHz Dual Core processor with 2GB RAM. We keep adaptive $\sum_{i=1}^n \omega_i = 1$ for all i flows, in which $\omega^{avionics} : \omega^{multimedia} : \omega^{data} = 6 : 3 : 1$ according to the observation of sampling-based approach in Section III-G.

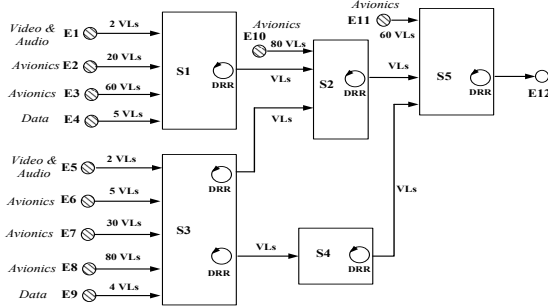


Fig. 6. AFDX prototype configuration.

Figure 6 shows the network configuration for our experiments, which include 11 source end systems, 1 destination end system and 5 switches. Specifically, we have 69 multimedia (video & audio) VLs, 335 avionics VLs and 152 best-effort data VLs, which are scheduled at each switch based on DRR policy. For comparison, we also implemented FIFO and static RMA policy. We use 16-bit values as virtual link IDs to route Ethernet frames in the entire AFDX network. A VL transmission ends when it passes through integrity and redundant checking as shown in Section III-E and the *SkewMax* here is set to 5ms.

TABLE I
PARAMETER SETTINGS FOR BAG AND FRAME LENGTH.

Avionics Flows			
BAG (ms)	Number of VLs	Frame Length (bytes)	Number of VLs
2	5	0-100	180
4	25	100-200	70
8	35	200-400	35
16	70	400-600	20
32	90	600-800	15
64	60	800-1000	10
128	50	> 1000	5
Multimedia Flows			
BAG (ms)	Number of VLs	Frame Length (bytes)	Number of VLs
2	2	0-100	0
4	2	100-200	1
8	5	200-400	5
16	10	400-600	8
32	24	600-800	12
64	16	800-1000	18
128	15	> 1000	25
Data Flows			
BAG (ms)	Number of VLs	Frame Length (bytes)	Number of VLs
2	2	0-100	5
4	2	100-200	8
8	6	200-400	16
16	25	400-600	35
32	60	600-800	50
64	32	800-1000	20
128	20	> 1000	18

In our simulation, we take into account three typical types of flows, i.e., avionics, multimedia (video & audio) and best-effort data, to comprehensively evaluate the proposed scheme in terms of end-to-end delays and fairness measure. Table I exhibits the BAG and frame sizes of avionics, multimedia and best-effort data flows, which come from a synthetic scenario to simulate a real-life AFDX network. All BAGs are limited within 128ms and most frames are smaller than 1000 bytes. All links run at 100Mbps. These parameters follow real-world environments [2], [19] where avionics VLs often display heavier workloads than multimedia and best-effort data flows.

We evaluate end-to-end delays of heterogeneous flows that are generated by following periodic and sporadic properties. We define the “periodicity” that indicates a period for transmitting a periodic message, and for a sporadic message, it is the minimal inter-arrival time between messages. The simulations set the periodicity to be 40ms, which actually serves as traffic shaping function. Therefore, for periodic flows with BAGs smaller than 40ms, we replay them every 40ms and others are replayed in the time interval of their own BAG values. Sporadic flows with burst arrive at switches according to a Zipfian distribution with parameter 0.85.

The simulations examine the end-to-end delays of variable loads that are represented as the number of transmitted VLs. Specifically, a VL is allowed to transmit when its BAG is larger than pre-defined BAG constraint. We examine the end-to-end delays according to the constrained BAG values from 2 to 128 with exponential growth of 2. The experimental results are the average values of 50 runs.

B. Experimental Results

We first present the end-to-end delays of heterogeneous flows, including avionics, multimedia and best-effort data, under three types of schedulers, i.e., FIFO, Static Priority (SP) and DRR, with the comparisons between theory and simulation

results. The utilization-based fairness measure is then used to show the relatively fairness in the DRR policy.

1) *End-to-end Delays*: Our experimental results are shown in Figure 7 and 8 respectively corresponding to periodical and sporadic flows when considering FIFO, SP and DRR scheduling policies. The end-to-end delays are evaluated with the increments of BAG values. Three scheduling policies produce different end-to-end delays due to their different strategies for heterogeneous flows. The DRR obtains the best performance since it can significantly decrease the traffic congestion and re-transmission that may produce extra delays.

We observe that the delay of FIFO is the largest while that of DRR is the smallest. FIFO produces over 3.2 times larger delay than DRR in the worst case. FIFO equally treats each flow that is served in its arrival order and can not efficiently support real-time transmission for heterogeneous flows, thus introducing the largest delays among three schedulers. SP, on the other hand, takes into account the flow priorities with static setting and a flow is thus allowed to transmit when the queues of all higher-priority flows are empty. However, it is hard for the SP policy to provide fairness to the transmitted flows since it ignores the differentiated frame sizes of different priorities, i.e., flow utilization, and introduces larger delays especially for relatively low-priority flows, possibly introducing data loss and high-cost re-transmission. In contrast, DRR utilizes round-robin fashion to fairly schedule incoming flows with better flexibility, thus obtaining the smallest end-to-end delays, and still maintain good fairness.

The end-to-end delays of heterogeneous flows including avionics, multimedia (video & audio) and best-effort data VLs, depend upon their frame sizes and scheduling policy. For instance, Figure 7 shows the delays of avionics flows compared with multimedia and data flows, when using FIFO, SP and DRR schedulers under periodic settings. We observe that avionics VLs have smaller delays than other two flows because performing the SP and DRR scheduling allows avionics flows to have higher priorities and more allocated quanta in each round, thus producing smaller delays. The results are also observed in the sporadic settings as shown in Figure 8.

Finally, the sporadic flows show larger delays than periodic flows since the former arrives following the Zipfian distribution that may produce more bursts and in contrast, the latter exhibits much smoother traffic. The observations conform to the conclusion in [20] and further prove the necessity of traffic shaping.

2) *Utilization-based Fairness Measure (UFM)*: Utilization is the resource consumption rate within certain time interval and can be used to evaluate the fairness of scheduling policies. We define the utilization fairness measure for each type as $UFM_{[0,t]}^x = \frac{\sum_{i \in x} \omega_i \Delta / U(x)_{[0,t]}}{\sum_{i \in x} \Delta / \sum U(x)_{[0,t]}}$, where $x \in \{\text{avionics}, \text{multimedia}, \text{data}\}$, $U(x)_{[0,t]}$ is the real transmitted frame length during time interval $[0, t]$. The equation actually shows the relative fairness of heterogeneous flows by examining their allocated quanta and real transmitted frame length compared with entire system during time interval $[0, t]$. The

variable t corresponds to the BAG in our simulation. The $UFM_{[0,t]}^x$ value is within interval $[0, 1]$ and when the value is close to 1, it means achieving more fairness.

Figure 9 shows the fairness measure based on actual utilization of heterogeneous flows when using DRR policy. It is observed that DRR can help all flows obtain relative fairness above 90%. In addition, periodic flows have better utilization fairness than sporadic flows since the latter experiences more bursts.

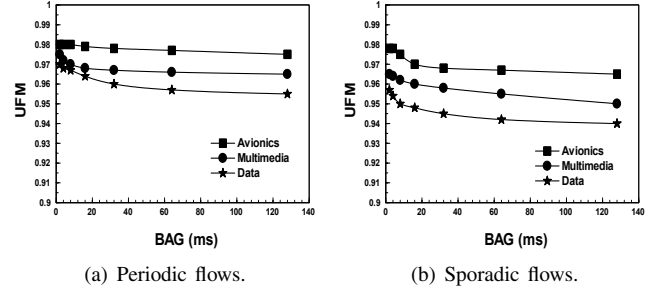


Fig. 9. Utilization-based fairness measures of heterogeneous flows.

V. RELATED WORK

AFDX Ethernet technology provides reliable deterministic network service and guaranteed bandwidth by using redundant virtual links. Recently, this technology has received much attention from industry and academic fields [2], [3].

Existing work mainly studies end-to-end delays based on the scheduling policies of FIFO, static priority and weighted fair queuing [19] for a single data flow. Three typical methods, including network calculus, queuing networks simulation and model checking, are compared in [3] for the evaluation of end-to-end delays when using FIFO scheduling in switch outputs. Deterministic network calculus often gives a guaranteed upper bound of end-to-end delay. Given an exceedable probability, authors in [21] take into account the transmission of three-type flows, including avionic, multimedia and best-effort data flows by utilizing the scheduling of FIFO and static priority.

Utilization-based schedulability analysis becomes an efficient tool for the design and implementations of real-time systems by deriving utilization bounds [22]. Utilization-bound schedulability analysis using Weighted Round Robin (WRR) [13] maximizes utilization bounds by establishing a unified modeling framework to facilitate the derivation of utilization bounds. Recently, non-utilization based schedulability analysis has been studied in [23], [24]. There still lacks the study of end-to-end delay analysis for heterogeneous flows, especially transmitted within independent networks for reliability concerns in the design of the AFDX network.

VI. CONCLUSION

An AFDX network provides reliable and deterministic delivery of frames for avionics applications by using redundant links that have bounded delays. This paper studies end-to-end delays of heterogeneous flows using DRR-based scheduling policy in an AFDX network. Different from existing state-of-the-art work, this paper focuses on studying the end-to-end delays of heterogeneous flows with different transmission

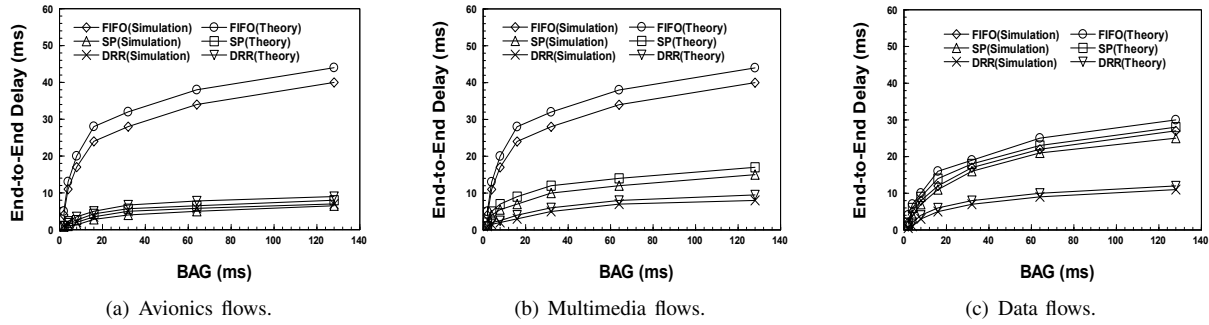


Fig. 7. End-to-end delays of periodical heterogeneous flows under FIFO, SP and DRR schedulers.

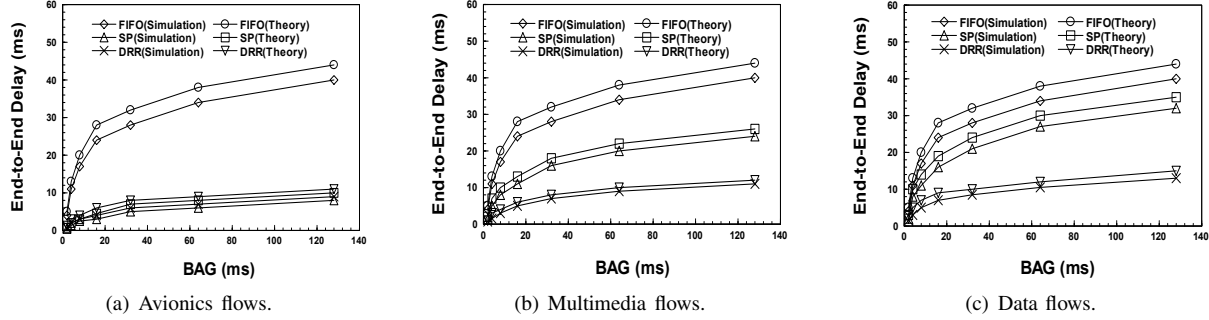


Fig. 8. End-to-end delays of sporadic heterogeneous flows under FIFO, SP and DRR schedulers.

requirements. We transform the problem of scheduling multiple flows into a new problem of quanta computation by using DRR-based policy to facilitate accurate analysis of end-to-end delays. We also investigate the effects of redundant management on end-to-end delays based on parameter analysis while supporting reliable transmission of avionics, multimedia and best-effort data flows. Extensive experimental results show the effectiveness and efficiency of our proposed scheme.

ACKNOWLEDGEMENT

This work was supported in part by University of Nebraska-Lincoln.

REFERENCES

- [1] H. Frazier and C. Inc, "The 802.3 z gigabit Ethernet standard," *IEEE Network*, vol. 12, no. 3, pp. 6–7, 1998.
- [2] ARINC 664, "Aircraft Data Network, Part 7 Avionics Full Duplex Switched Ethernet (AFDX) Network," ARINC 05-005/ADN-39, 2005.
- [3] H. Charara, J. Scharbarg, J. Ermont, and C. Fraboul, "Methods for bounding end-to-end delays on an AFDX network," *Proc. ECRTS*, 2006.
- [4] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking (ToN)*, vol. 6, no. 5, p. 624, 1998.
- [5] P. Brucker, "Scheduling algorithms," *Springer Verlag*, 2007.
- [6] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on networking*, vol. 4, no. 3, pp. 375–385, 1996.
- [7] L. Lenzi, E. Mingozzi, and G. Stea, "Tradeoffs between low complexity, low latency, and fairness with deficit round-robin schedulers," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 681–693, 2004.
- [8] H. Chaskar and U. Madhoo, "Fair scheduling with tunable latency: a round-robin approach," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 592–601, 2003.
- [9] R. Cruz, "A calculus for network delay. I. Network elements in isolation," *IEEE transactions on information theory*, vol. 37, no. 1, pp. 114–131, 1991.
- [10] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.
- [11] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1265–1279, 1991.
- [12] Y. Hua, "Fair scheduling scheme with feedback in the joint allocation of heterogeneous resources," *Proc. ICC*, 2006.
- [13] J. Wu, J. Liu, and W. Zhao, "Utilization-bound based schedulability analysis of weighted round robin schedulers," *Proc. RTSS*, pp. 435–446, 2007.
- [14] Y. Jenq, "Performance analysis of a packet switch based on single-buffered banyan network," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 6, pp. 1014–1021, 1983.
- [15] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 611–624, 1998.
- [16] L. Rizzo, "Dummysnet: a simple approach to the evaluation of network protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 1, pp. 31–41, 1997.
- [17] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostić, J. Chase, and D. Becker, "Scalability and accuracy in a large-scale network emulator," *Proc. OSDI*, pp. 271–284, 2002.
- [18] M. Bramson, "Instability of FIFO queueing networks with quick service times," *The Annals of Applied Probability*, pp. 693–718, 1994.
- [19] J.-L. Scharbarg, F. Ridouard, and C. Fraboul, "A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 38–49, February, 2009.
- [20] A. Mifdaoui, F. Frances, and C. Fraboul, "Full Duplex Switched Ethernet for Next Generation "1553B"-based Applications," *Proc. RTAS*, 2007.
- [21] F. Ridouard, J. Scharbarg, and C. Fraboul, "Probabilistic upper bounds for heterogeneous flows using a static priority queueing on an AFDX network," *Proc. IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1220–1227, 2008.
- [22] J. Wu, J. Liu, and W. Zhao, "On schedulability bounds of static priority schedulers," *Proc. RTAS*, pp. 529–540, 2005.
- [23] X. Liu and T. Abdelzaher, "On non-utilization bounds for arbitrary fixed priority policies," *Proc. of the Twelfth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 167–178, 2006.
- [24] X. Liu and T. Abdelzaher, "Non-utilization bounds and feasible regions for arbitrary fixed-priority policies," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 10, 2010.