IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 22, 2016, accepted June 16, 2016, date of publication June 24, 2016, date of current version October 6, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2584859

System-Level Scheduling of Mixed-Criticality Traffics in Avionics Networks

JIANGUO YAO[1,2,4], JIAHONG WU[2], QINGCHUN LIU[1], ZHIYONG XIONG[1], AND GUCHUAN ZHU[3]

[1]Science and Technology on Avionics Integration Laboratory, Shanghai 200233, China
[2]Shanghai Jiao Tong University, Shanghai 200240, China
[3]École Polytechnique de Montréal, Montréal, QC H3T 1J4, Canada
[4]Department of Informatics, Technische University München, München 85748, Germany

Corresponding author: J. Yao (jianguo.yao@sjtu.edu.cn)

This work was supported by the Aeronautical Science Foundation of China within the Science and Technology on Avionics Integration Laboratory of China under Grant 20145557010.

ABSTRACT System-level mixed-criticality design aims at reducing production cost and enhancing resource efficiency. This paper studies the technology of integrating mixed-criticality avionics traffics for Avionics Full-Duplex Switched Ethernet (AFDX) network, which can transmit both critical and non-critical traffics. These two traffics have different QoS requirements, such as low latency for critical traffics and high bandwidth for non-critical traffics. We use system-level compositional scheduling to integrate mixed-criticality traffics into one network to enhance the scalability of AFDX network. In the architecture of the proposed compositional scheduling, critical traffics are scheduled by bandwidth allocation gap-based scheduler, and non-critical traffics by Round Robin manner. To estimate the delay bound meeting requirements of applications, end-to-end delay for both critical and non-critical traffics are analyzed by using network calculus. Finally, a true time-based simulation of AFDX networks is conducted to verify the effectiveness of the proposed approach.

INDEX TERMS Avionics full-duplex switched ethernet (AFDX), mixed-criticality, end-to-end delay, network calculus.

I. INTRODUCTION

System-level designing should be so done as to provide different services that meet the given requirements of specific applications. Traditionally, two approaches are adopted for engineering designs [1], namely critical engineering and best-effort engineering, which are widely used to facilitate implementation of various applications. Critical engineering design is based on the worst-case scenario to meet the requirements in dangerous environments; best-effort engineering design is based on the average-case scenario and dynamic resource management. As the processing speed and resource capacity increase, modern system-level designs prefer to handle mixed criticality applications to reduce cost and enhance systemaŕs scalability. Handling mixed criticality applications is a concept that allows applications of different levels of criticality to interact and co-exist on the same platform. This paper will study the mixed criticality integration technology for avionics systems of commercial airplanes.

Avionics Full-Duplex Switched Ethernet (AFDX) network has been developed for supporting critical avionics applications [2], such as flight control system, navigation system, and engineering control system. Diverse problems relating to these applications have been extensively studied, and the most representative among them include the following: (i) how to schedule and estimate the delay of end system for time-critical applications, (ii) how to tighten the bound of End-to-End delay for AFDX network etc [3]. Some problems relating to system-level resource management and optimization, however, remain unresolved. For example, experimental studies show that, for providing guaranteed determinist to high level criticality, such networks have to be lightly loaded [4], using typically not more than 20% of the bandwidth that the physical link can provide. On the other hand, the same network is also expected to support non-critical traffics relating to file transmission, data backup, system configuration etc [5]. Providing support to mixed criticality traffics is theoretically challenging, but practically useful. In the present work, the emphasis is on system-level scheduling and End-to-End delay analysis of both critical and non-critical traffics [3].

Virtual Link (VLs) is a major concept of the AFDX network environment and it is responsible for routing data flow

and controlling data transmission rate. End systems (ESs), which are mutually linked with limited switches and physical connections, differ from physical links. The VL is a bidirectional logic path, based on the physical links; it links one ES to one or more ESs. VLs are rate limited whose dedicated bandwidths are decided by the two parameters, Bandwidth Allocation Gap (BAG) and the largest frame length $L_{\max}$. Ranging from 1 to 128 milliseconds in powers of 2, the values of BAG provide limited dynamical range for flow control. Furthermore, BAG is application-specific and is suitable for the role of scheduling parameter in AFDX network [6].

One of the challenges in implementing system-level optimization for mixed-criticality traffics is providing low-latency to critical traffics, and simple, best-effort traffic guarantees to non-critical traffics. To this end, the compositional scheduling is utilized to serve different types of traffics in different ways. In the scheduler, critical traffics are scheduled by Bandwidth Allocation Gap (BAG) based scheduler and non-critical traffics by Round Robin manner [3]. The main contributions of this paper are three-fold:

- We propose the feasible system integration architecture, using compositional scheduling for AFDX networks mixed-criticality traffics.
- We exploit network calculus to analyze End-to-End worst-case delay for both critical and non-critical traffics.
- We demonstrate how to implement the proposed architecture, while ensuring latency-constraints for critical traffics, and high bandwidth for non-critical traffics.

The remainder of the paper is organized as follows: Section II summarizes the related works; Section III discusses the system architecture, proposed for integrating mixed-criticality traffics in avionics networks; Section IV presents the End-to-End delay analysis for mixed-criticality traffics in AFDX network; Section V verifies the proposed approach by simulating a Truetime based AFDX network; and Section VI presents the conclusion drawn from this study.

II. RELATED WORK

In this section, we briefly summarize the existing works on AFDX networks and compositional scheduling, and highlight the novelty of this paper.

AFDX becomes a research hotspot soon after the first release of the network standard [2]. It has been initially adopted by Airbus Industries for their A380 and A350 civil airplanes, and later by Boeing for their Boeing 787 Dreamliner. A lot of work on analysis of network scheduling and End-to-End delays in AFDX network was carried out thereafter.

Gutiérrez et al. propose feasible methods for the analysis of response-time of network distribution with deterministic switched in AFDX [7]. Focusing on the designing strategies of scheduling methods to reduce End-Systems level transmission jitter in AFDX network, [8] optimizes Bandwidth Allocation Gap and frame length size in virtual links, which can address the problem of transmission jitter with a new scheduling policy. Tawk et al. describe a method to schedule and estimate the delay of end system for time-critical applications [6]. To accurately calculate the worst-case latency of a given data flow in a determinate routing path, Charara et al. present a limited model-checking approach, based on timed automata [9], but it is useful only in small example scenarios. The technique, based on mixed AFDX/CAN architecture, presents some functions that are necessary in the gateway nodes to satisfy the End-to-End delay and to guarantee End-to-End real-time communication [10]–[12], etc.

To analyze the End-to-End delays of all the transmitted flows of AFDX network, Georges Kemayo et al. propose a method, which focuses on Forward End-to-End delay Analysis (FA) to reduce the pessimism of all the computed bounds, thus facilitating the estimation of the End-to-End delay more accurately [13]. Moreover, the task of improving the reliability of the accuracy of End-to-End delay estimation is complicated. Based on the SNC theory, Zhitao Wu et al. build a stochastic traffic envelope function and propose a new probabilistic algorithm for a better approximation of the estimation of AFDX packages transmission delay [14]. However, for calculating the flow-based End-to-End delay in large networks, the methods mentioned above require great effort to capture cross-traffic multiplexing and worst-case flow scheduling throughout the network. Similarly, to derive network calculus delay bounds, Steffen Bondorf et al. also continue to contribute to the algebraical approach that is feasible for large networks. They focus on the feed-forward network and bound the arrival of cross-traffics at the same locations which as result cause interference to analysis flow tandem of end servers [15].

Furthermore, the algorithm adopted for performing End-to-End delay analysis of real-time networking or related system is based on compositional scheduling. It has become one of the most popular resource management scheduling algorithms for solving the problem of various time-critical systems [16]–[18].

All the above mentioned works discuss only one kind of AFDX package (critical traffics or non-critical traffics). To the best of our knowledge, our present study is the first attempt to integrate mixed-criticality traffics. In this work, we study End-to-End delay analysis and ensure that non-critical traffics are allocated more bandwidth, and the critical traffics have satisfying latency in max-critical AFDX networks. Completely different from the previous researches, our work is specifically targeted to AFDX network, because none of the previous works can be directly applied to AFDX networks.

III. MIXED-CRITICALITY TRAFFICS INTEGRATION ARCHITECTURE

A. MIXED-CRITICALITY TRAFFICS IN INTEGRATED MODULAR AVIONICS NETWORKS

The Airplane is partitioned into a number of functional domains [5]: cockpit domain, cabin domain, energy domain,

and utilities domain. In the airplane, the AFDX network and the Integrated Modular Avionics (IMA) computing modules, constituting the core of the avionics system, are split. The partitioned systems in IMA computing modules are typical safety-critical applications, such as generator control units, flight control systems, and main engine Full Authority Digital Engine Control units (FADECs); they also include applications relating to non-critical traffics, such as simultaneous file transmission and video monitoring.

B. INTEGRATING MIXED CRITICALITY TRAFFICS IN DIFFSERV-BASED ARCHITECTURE

In the current airplane, such as A380, critical and non-critical traffics exist simulatively, as mentioned in the last subsection. As AFDX standard is developed to support real-time safety-critical applications, integration of all the traffics is of utmost importance. The AFDX network architecture may integrate both the traffics by DiffServ (Differentiated Services)-based QoS management mechanism [19], which employs a fixed priority scheduling method to serve the applications with different priorities. DiffServ-based architecture can provide low-latency to critical traffic, and simple, best-effort traffic guarantees to non-critical traffic. It has been shown that the networks using DiffServ architecture can guarantee bounded End-to-End delay for traffics with the highest priority, regardless of the level of network utilization [20]. Usually, DiffServ uses a 6-bit Differentiated Services Code Point (DSCP) field in the header of IP packets for packet classification.

DiffServ-based QoS management mechanism consists of two layers, namely critical layer and non-critical layer. The critical layer manages high priority critical applications for which real-time and reliable transmission is ensured. The non-critical layer manages low priority non-critical applications, which may require a larger data rate (throughput), but less stringent adherence to real-time requirements, as compared to the requirements of critical layer. In this mechanism, non-critical applications will be served only when there are no critical traffics. The flow in non-critical layer has little impact on the critical layer, as it avoids disturbing the critical applications. Besides, the critical layer can transmit packets through the other layer to gain reliability and real-time assurance. The End-to-End delay for critical traffics, the most important feature of QoS measurements, will be analyzed in this paper by employing Network Calculus. A major issue discussed in this paper is judicious utilization of the bandwidth to avoid congestion, because critical applications require only about 20% bandwidth. To address this issue, optimal bandwidth allocation technology is employed for AFDX networks.

C. Diff Serv-BASED COMPOSITIONAL SCHEDULING

We use compositional scheduling frame-work to manage mixed-criticality traffics integration. Several compositional frameworks, each designed with its own scheduling scheme [3], are available, but we have chosen the combined BAG-based and Round Robin compositional scheduler, as shown in Fig. 1.

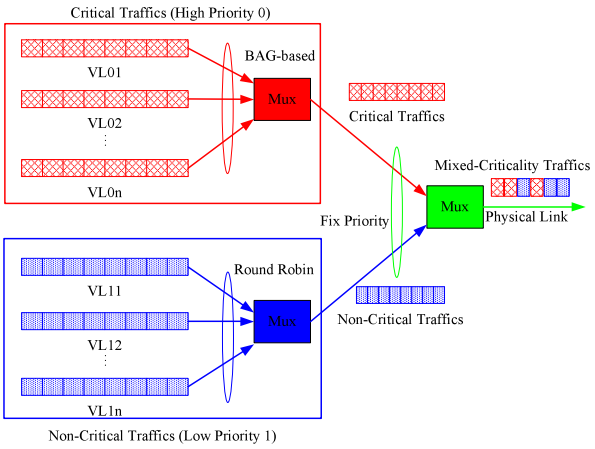


FIGURE 1. Compositional scheduling for mixed-criticality traffics.

The traffics from different VLs are aggregated into the same class, using BAG-based scheduler for critical traffic and Round Robin manner for non-critical traffic. Two classes, namely high priority critical class and low priority non-critical class, are scheduled by fix priority manner.

D. FIX PRIORITY SCHEDULING BETWEEN CRITICAL TRAFFICS AND NON-CRITICAL TRAFFICS

The major problem of integrating critical and non-critical traffics is to separate one from the other and decrease the impact of non-critical traffics on critical traffics. To this end, we use compositional scheduling architecture, based on Fix Priority scheduling, which consists of two mechanisms: non-preemptive scheduling and preemptive scheduling.

Non-preemptive scheduling is usually considered inferior to preemptive scheduling for time or/and safety critical systems, because it leads to poor task responsiveness. But, it has certain advantages, like low implementation complexity and run-time overhead. Hence, non-preemptive scheduling is considered better than preemptive scheduling for many real-time applications on engineering AFDX platform. In this paper, we adopt Non-preemptive Fix Priority scheduling to integrate critical and non-critical traffics.

E. BAG-BASED SCHEDULING FOR CRITICAL TRAFFICS

AFDX networks are designed to provide deterministic, real-time and reliable transmission service. Virtual link (VL) is a typical feature in the configuration of AFDX networks, whose routing mechanism, using VLs, is static. In each VL, a regulator is employed for allocation of bandwidth and no more than one packet is sent during each interval of BAG. To this end, the well known leaky bucket is used by AFDX specification, which is characterized as two main paraments $\sigma_i = l_i^{max}$ and $\rho_i = L_i^{max}/BAG_i$, where L_i^{max} and BAG_i are the maximum frame length and BAG for VL i respectively. All the VLs in AFDX network conform to the leaky bucket constraints, and the delay upper bound for each VL can be guaranteed, even with simple scheduling strategies, such as

the First-In-First-Out (FIFO) strategy. However, this mechanism ignores the fact that Mixed-Criticality traffics require different transmission QoS. The FIFO buffer thus introduces potential transmission bottleneck from concurrent arrival of packages due to queuing delays in switches. Inefficiency in scheduling Mixed-Criticality traffics that are aggregated in the FIFO buffer also increases the delay. To efficiently manage both critical and non-critical traffics, we propose compositional DiffServ-Alike Scheduling, which guarantees different QoS requirements, such as tight delay bound for critical traffics and high throughput for non-critical traffics.

To avoid missing deadline (assuming as the BAG), the schedulers must ensure that every VL should be exactly served before the successive frame come alone in the same VL. To this end, the schedulability is usually verified whether the deadline is missed, i.e. the wort-case End-to-End transmission delay is smaller than the BAG.

To avoid missing deadline (assuming as the BAG), the schedulers must ensure that every VL should be exactly served before the successive frame come along in the same VL. To this end, the schedulability is usually verified by checking whether the deadline is missed, i.e. whether the wort-case End-to-End transmission delay is smaller than the BAG. To manage the VLs of critical traffics, we employ the BAG-based scheduling policy, according to which, the smaller the BAG, the higher is the priority. It is to be noted that packets belonging to the same VL arrive with a period of BAG, and that the BAG-based scheduling policy is an implementation of the well-known Rate-Monotonic Algorithm (RMA) [21]–[23]. The schedulability condition for a single scheduling has been given as [6]:

$$\sum_{j=1}^{i-1} \frac{L^0_{j,\max}}{c} \lceil \frac{BAG^0_i}{BAG^0_j} \rceil + \frac{\max_{j>i}\{L^0_{j,\max}\} + L^0_{i,\max}}{c} \leq BAG^0_i, \quad (1)$$

where c is the maximum transmission rate of a physical link.

In the proposed compositional scheduling, the non-critical traffics with low priority will have impact on the critical traffics, and one of the frames may be blocked by non-critical traffics. Hence the new schedulability condition for compositional scheduling is as follows:

$$\sum_{j=1}^{i-1} \frac{L^0_{j,\max}}{C} \lceil \frac{BAG^0_i}{BAG^0_j} \rceil + \frac{\max_{j>i}\{L^0_{j,\max}\} + L^1_{\max} + L^0_{i,\max}}{C} \leq BAG^0_i, \quad (2)$$

where $L^1_{\max}$ is the maximum frame length of non-critical traffics.

F. ROUND ROBIN SCHEDULING FOR NON-CRITICAL TRAFFICS

To provide the best-effort throughput and to ensure fairness of non-critical traffics scheduling mechanism in AFDX networks, we manage their VLs by adopting Round Robin scheduling. This is an inefficient, but an alternative, approach for first-come first-served queuing in situations wherein the best-effort packet switch or other statistical multiplexing is to be handled.

In the Round Robin scheduling mechanism, every VL can be identified uniquely by the source and destination addresses. All or some of the VLs hold a multiplexer, which is responsible for separating the transmission queue of each VL, following the method mentioned above. The advantage of Round Robin scheduling is that it can help isolate active VLs, while transferring packages on a shared channel, besides separating the channels in a periodically repeated transmission order. The scheduling involves load balancing and high utilization rate, because it tries to prevent link resources from going to be unused while taking place by the out of packet to the waiting data flow.

The max-min fairness of Round-Robin scheduling depends on the equality of data packet sizes. The packets of maximum length always require the longest time, because of which the VL must await longer for the completion of such transmission. So, too much variation in the length of data packets, between one transmission job and the other, may not be desirable.

IV. DELAY ANALYSIS FOR MIXED-CRITICALITY TRAFFICS

A. SINGLE-HOP MULTIPLEXING DELAY ANALYSIS

Network calculus is used for theoretical analysis of the performance of computer networks, and it is based on Min-plus and Max-plus Calculus [24]. Using convolution formula as a tool of Network calculus and employing the "pay-bursts-only-once" property in the analysis, the End-to-End delay bound of network transmission can be derived.

Theorem 1 (Critical Traffics Class Delay): Consider n_0 non-preemptive flows with high priority in critical traffics class. The arrival curve for each flow conforms to (σ,ρ) model, which is $\alpha^0_j(t) = \frac{L^0_{j,\max}}{BAG^0_j} \cdot t + L^0_{j,\max}$, where 0 denotes the critical traffics, and j the jth flow of critical traffics class. The delay of flow j is

$$D^c_j = \frac{L^0_{j,\max}}{C^0_j} + T^0_j. \quad (3)$$

Theorem 2 (Non-Critical Traffics Class Delay): Consider n_1 non-preemptive flows with high priority in non-critical traffics class. The arrival curve for each flow conforms to (σ,ρ) model, which is $\alpha^1_j = \frac{L^1_{j,max}}{BAG^1_j} \cdot t + L^1_{j,max}$, where 1 denotes the non-critical traffics, and j the jth flow of non-critical traffics class. The delay of flow j is

$$D^n_j = \frac{L^1_{j,\max}}{C^1} + T^1_j. \quad (4)$$

B. MULTI-HOP END-TO-END DELAY ANALYSIS

In AFDX networks, the End-to-End delay of one VL subject to source End System, Switches, destination End System and

propagation in physical links [3]. It can be characterized by

$$D_{VL} = D^s_{ES} + D_{switches} + D^d_{ES} + D_p, \quad (5)$$

where D^s_{ES} and D^d_{ES} are the delays that occurred at source End System and destination End System respectively, and $D_{switches}$ are the delays spent in switches, D_p is the propagation delay in physical links, which easily conforms to an upper bound T^p_{max} for each VL [3].

To obtain the multi-hop AFDX End-to-End delay bound, by using compositional scheduler for each VL, we separately studied each part of End-to-End delay, based on the above composition (except for propagation delay).

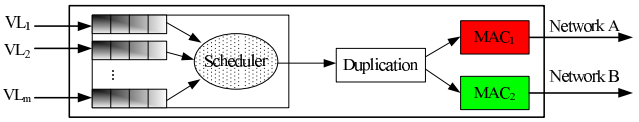


FIGURE 2. Source end system architecture.

1) DELAY FROM SOURCE END SYSTEM

The Source End System, whose architecture is shown in Fig. 2, regulates transmitted data to each VL, based on traffic shaping function. . The scheduler passes the VLs from input buffers to duplication modular. Then, two duplications are transmitted to different networks. The delays produced in the source End System are denoted by D^s_{ES}.

Definition 1 (Delay of Source End System): The Delay offered by the *j*th flow of critical or non-critical class in the Source End System is given by

$$D^s_{ES}(j) = D_j + D_{dup}, \quad (6)$$

where D_j is delay produced for critical or non-critical flows in the scheduler, and D_{dup} is duplication delay for Network A or Network B [3].

Proof: The end system has two sources to produce the delay: scheduler and duplication. Hence, the above theorem is proven. ■

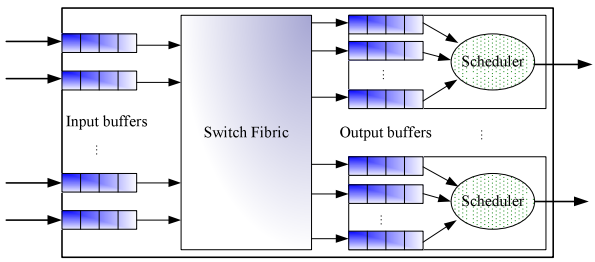


FIGURE 3. The architecture of AFDX switch in DiffServ-Alike mechanism.

2) DELAY FROM SWITCHES

The architecture of AFDX Switch, shown in Fig. 3, has mainly three parts: input buffer, forwarding mechanism and compositional scheduler. Upon receiving the packets from physical link, they are buffered in the input queue, which produces a delay depending on the size of the frame length. Forwarding mechanism gets the packets from the input buffer and puts them in the corresponding destination output queue by looking up static forwarding table, and this introduces a constant delay. Compositional scheduler sends the packets to physical links, which produces the queuing delays.

Definition 2 (Delay of a Switch): The Delay offered by the *j*th flow of critical or non-critical class in one Switch is given by

$$D^s_{ES}(j) = D^{switch}_j + D_{forward}, \quad (7)$$

where D^{switch}_j is the delay produced by the scheduler for critical or non-critical flows in a switch, and $D_{forward}$ is constant delay of the forwarding mechanism [3].

Proof: A switch has two sources to produce the delay: scheduler and forwarding mechanism. Hence, the above theorem is proven. ■

3) DELAY FROM DESTINATION END SYSTEM

In the Destination End System, two mechanisms are employed: integrity checking and redundancy management, as shown in Fig. 4. Integrity checking is used to guarantee the accuracy of received frames. Under fault-free network mode, the Integrity Checking just passes the received frames to the Redundancy Management for independent network. If fault network mode is employed, Integrity Checking discards the invalid frames, based on sequence number, and then informs the network management of the invalid frame. The Redundancy Management handles two types of VLs, redundant VLs and non-redundant VLs, each in a different way. The redundant VLs are sent through both A and B networks and the non-redundant VLs, that is the frames, only through network A or B. In the Redundancy Management, *SkewMax* is used to define the maximum time period during which the same two valid frames are received from both the Networks A and B. We use D^d_{ES} to indicate the processing delay in destination End System.

Definition 3 (Delay of Destination End System): The Delay offered by the *j*th flow of critical or non-critical class in the Destination End System is given by

$$D^d_{ES}(j) = D_{IC} + D_{RM}, \quad (8)$$

where D_{IC} is Integrity Checking delay, and D_{RM} is Redundant Management delay [3].

Proof: The Destination End System has two sources to produce the delay: scheduler, and Integrity Checking and Redundancy Management. Hence, the above theorem is proven. ■

4) END-TO-END DELAY

We study the multi-hop delay from multiple switches by using the ''pay-bursts-only-once'' property of network calculus [24]. To understand the phenomenon of ''pay-bursts-only-once'', we introduce the following theorem of concatenation of switches.

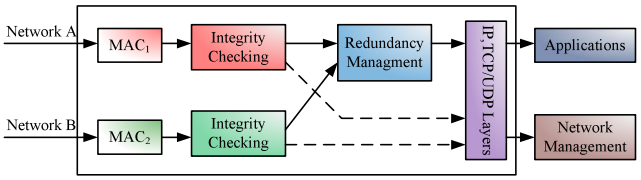


FIGURE 4. Destination end system.

Theorem 3 (Concatenation of Nodes [24]): Assume that a flow of VL traverses two nodes in sequence with service curves $\beta_1(t)$ and $\beta_2(t)$. Then, the concatenation of the two nodes offers a service curve of $\beta(t) = (\beta_1 \otimes \beta_2)(t)$, like that of a single node.

This theorem is easily proven by citing the association of min-plus convolution, i.e., $((R \otimes \beta_1) \otimes \beta_2)(t) = (R \otimes (\beta_1 \otimes \beta_2))(t)$. From this theorem, we can see that the ''pay-bursts- only-once'' phenomenon, in fact, allows the End-to-End delays and backlog bounds to be scaled-up with increasing number of multiple nodes, rather than quadratic scaling in summing up single node.

The transmission time and the aggregate waiting time in queues are the two main factors that affect the accuracy of End-to-End delay analysis. Network calculus can help correlate these factors tightly and make the upper bound of the End-to-End delay more accurate. So, the network calculus is suitable for delay analysis of elementary entities in the AFDX network.

Corollary 1 (Concatenation of Switches [24]): Consider a flow of VL that traverses through a set of switches $S = S_1, S_2, \cdots, S_m$ whose service curves are $\beta_j, (j = 1, \cdots, m)$. The switches can be concatenated into a single switch by using the convolution of nodes theorem, after which the End-to-End service curve will be $\beta(t) = (\beta_1 \otimes \beta_2 \otimes \cdots \otimes \beta_m)(t)$, like that of a single node.

Theorem 4 (End-to-End Service Curve): The End-to-End service curve for the jth flow in the first switch is given by

$$\beta_j^{e2e}(t) = \min_{1\le k\le n}\left[C_j(k)\right]\left(t - \sum_{k=1}^{n} T_j(k)\right), \tag{9}$$

where n is the number of hops, and $C_j(k)$ and $T_j(k)$ are service rate and queuing delay of VL j in switch k, respectively.

Proof: The End-to-End service curve is defined as $\beta^{e2e}(t) = R(t - T_s^t)$, where T_s^t denotes the total queuing delay. The worst case total queuing delay of the jth flow can be represented by $T_s^t = \sum_{k=1}^{n} T_j(k)$. ■

Corollary 2 (Critical Traffics Class End-to-End Service Curve [3]): The End-to-End service curve offered by the jth flow of critical class in the first switch is given by

$$\beta_{0j}^{e2e}(t) = \min_{1\le k\le n}\left[C_j^0(k)\right]\left(t - \sum_{k=1}^{n} T_j^0(k)\right). \tag{10}$$

Corollary 3 (Non-Critical Traffics Class End-to-End Service Curve [3]): The End-to-End service curve offered by the jth flow of non-critical class in the first switch is given by

$$\beta_{1j}^{e2e}(t) = \min_{1\le k\le n}\left[C^1(k)\right]\left(t - \sum_{k=1}^{n} T_j^1(k)\right). \tag{11}$$

Theorem 5 (Critical Traffics Class End-to-End Delay [3]): The End-to-End Delay offered by the jth flow of critical class in the first switch is given by

$$D_{0j}^{e2e} = \frac{L_{j,max}^0}{\min\limits_{1\le k\le n}\left[C_j^0(k)\right]} + \sum_{k=1}^{n} T_j^0(k) + n\frac{L_{j,max}^0}{R} + D_{dup} + n\cdot T_f + D_{IC} + D_{RM}. \tag{12}$$

Proof: According to the theorem 1.4.3 [24], given a flow with arrival curve $\alpha(t)$ and service curve $\beta(t)$, the delay of the flow bounded by the maximum horizontal deviation between the arrival curve α and the service curve β, is represented by $h(\alpha, \beta) = \sup_{s\ge 0}(\inf\{\tau \ge 0 | \alpha(s) \le \beta(s+\tau)\})$. Then using the service curve (10) and the arrival curve (σ,ρ) model $\alpha_j^0(t) = \frac{L_{j,\max}^0}{BAG_j^0}\cdot t + L_{j,\max}^0$, we get the delay of flow j as $\frac{L_{j,max}^0}{\min\limits_{1\le k\le n}\left[C_j^0(k)\right]} + \sum_{k=1}^{n} T_j^0(k)$). Considering the duplication delay, forwarding delay, Integrity checking delay and Redundancy Management delay, we then derive the actual End-to-End delay of critical traffics as given by (12). ■

Theorem 6 (Non-Critical Traffics Class End-to-End Delay): The End-to-End Delay offered by the jth flow of non-critical class in the first switch is given by

$$D_{1j}^{e2e} = \frac{L_{j,max}^1}{\min\limits_{1\le k\le n}\left[C^1(k)\right]} + \sum_{k=1}^{n} T_j^1(k) + n\frac{L_{j,max}^1}{R} + D_{dup} + n\cdot T_f + D_{IC} + D_{RM}. \tag{13}$$

Proof: Just as in the case of proof of Theorem. 2, we get the End-to-End Delay with perfect transmission as $\frac{L_{j,max}^1}{\min\limits_{1\le k\le n}\left[C^1(k)\right]} + \sum_{k=1}^{n} T_j^1(k)$. By considering the duplication delay, forwarding delay, Integrity checking delay and Redundancy Management delay, we finally get the actual End-to-End Delay of non-critical traffics as shown in (13). ■

5) DISCUSSIONS

The delay analysis presented above for mixed criticality traffics in AFDX network shows that the worst case End-to-End delay of critical traffics is impacted by non-critical traffics through its only one maximum frame length, as shown in (3) and (12). Similarly, the worst case End-to-End delay of non-critical traffics is also impacted by the critical traffics through its occupied bandwidth, as shown in (2) and (13). It is to be noted that maximum frame length and BAG, the parameters of occupied bandwidth, also impact the delay of critical traffics.

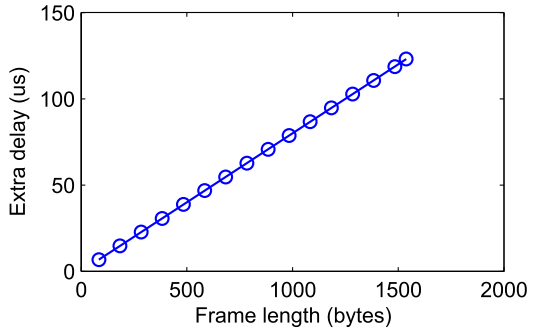


FIGURE 5. The extra delay blocked by non-critical traffics.

For mixed criticality traffics integration, more emphasis is put on the impact of non-critical traffics on critical traffics. From (3), we can easily deduce the main delay of critical traffics, not the non-critical traffics, because there is only one frame block for non-preemptive scheduling. The impact of non-critical traffics can be decreased in two ways: preemptive scheduling (only critical traffics can preempt non-critical traffics) and decreasing the maximum frame length of non-critical traffics with non-preemptive scheduling. By using preemptive scheduling, critical traffics can be made independent from non-critical traffics, but it results in losing the packet of non-critical traffics. Reliable packet delivery requires a re-transmission mechanism, which complicates the network architecture with additional bandwidth overhead. For this study, we use the second option of decreasing the maximum frame length of non-critical traffics with non-preemptive scheduling. The extra delay blocked by non-critical traffics is shown in Fig. 5. For implementing mixed criticality traffics integration, two delays are evaluated to ascertain if the non-critical traffics can be imported into the AFDX network: the End-to-End worst delay of critical traffics with the smallest frame blocking non-critical traffics D_c and the maximum acceptable End- to-End delay of critical applications $\bar{D}_c$. Then, we have the following proposition.

Proposition 1: If $D_c < \bar{D}_c$, the non-crtical traffics can be intergrated into the AFDX network, and the maximum frame length of non-critical traffics is determined by $64 + (\bar{D}_c - D_c)/nc$, where n is the number of hops and c is the physical rate of AFDX networks (100Mbps or 1000Mbps).

Corollary 4 (End-to-End Delay): The upper bound of the End-to-End delay of a flow being transmitted through an AFDX network is

$$E(D_{VL}) = \begin{cases} D_{VL}(A), non - redundant(A) \\ D_{VL}(B), non - redundant(B) \\ D_{VL}(A) + SkewMax, redundant(A) \\ D_{VL}(B) + SkewMax, redundant(B) \end{cases} \quad (14)$$

where $D_{VL}(A)$ and $D_{VL}(B)$ denote, respectively, the delays of original and redundant VL flows of Networks A and B. *non-redundant(A)* means that the VL is non-redundant and is sent by Network A; *non-redundant(B)* means that the VL is non-redundant and is sent by Network B. *redundant(A)* means that the VL is redundant and the frame comes first in Network A; *redundant(B)* means that the VL is redundant and the frame comes first in Network B.

V. SIMULATIONS AND EVALUATIONS

To verify the effectiveness of the proposed integrated mixed-criticality architecture for AFDX network, extensive simulations are conducted. The simulation results are summarized and analyzed in the following section.

A. SIMULATION SETUP

The simulations for this study are carried out by the platform of Real-time network control system simulation software, called True-Time–a Matlab/Simulink-based simulator [3]. The simulation configuration is depicted in Fig. 6. Critical traffic and non-critical traffic are multiplexed, but distinct safe-critical or priority of traffics holds different BAG and rate-limit which specified in Table. 1 and Table. 2.

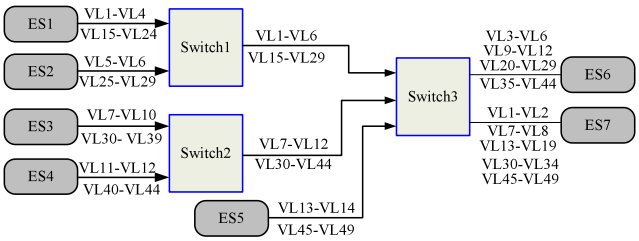


FIGURE 6. Simulation configuration of AFDX network.

TABLE 1. The parameters of critical VLS.

VL No.	1	2	3	4	5	6	7
BAG(ms)	16	4	2	16	4	32	4
$L_{\max}$(Bytes)	300	600	100	800	100	400	300
VL No.	8	9	10	11	12	13	14
BAG(ms)	16	16	8	64	16	128	16
$L_{\max}$(Bytes)	600	400	900	200	500	800	1000

TABLE 2. The parameters of non-critical VLS.

VL No.	15	16	17	18	19	20	21
BAG(ms)	4	32	64	128	2	16	32
VL No.	22	23	24	25	26	27	28
BAG(ms)	2	16	128	32	2	16	128
VL No.	29	30	31	32	33	34	35
BAG(ms)	32	4	16	128	64	4	1
VL No.	36	37	38	39	40	41	42
BAG(ms)	128	32	16	64	32	1	32
VL No.	43	44	45	46	47	48	49
BAG(ms)	64	128	2	1	64	128	16

This simulator includes mainly two interface modules: TrueTime Kernel and TrueTime Network. The simulator addressees the co-simulation of controller task by TrueTime Kernel, and the network transmissions by TrueTime Network. Independent tasks, periodic or non-periodic, can be simulated in TrueTime Kernel. TrueTime Network can simulate various network protocols, including CSMA/CD(Ethernet),

CSMA/AMP(CAN), Switched Ethernet, TDMA, FDMA, and Round Robin. It can also simulate the Wireless networks (802.11b/g WLAN and 802.15.4 ZigBee) and battery-powered devices. In our simulation for AFDX networks, usually, ES is responsible for generating VLs to simulate periodic tasks, which can be handled by the TrueTime Kernel. AFDX switch, being the core of the network protocol, can perform in the TrueTime Network module. And, SP scheduling is added to the TrueTime Network module by modifying the source TrueTime code.

B. END-TO-END DELAY EVALUATIONS

While evaluating the simulation, the transmission delay of critical VLs are observed for criticality architecture only and mixed criticality architecture.

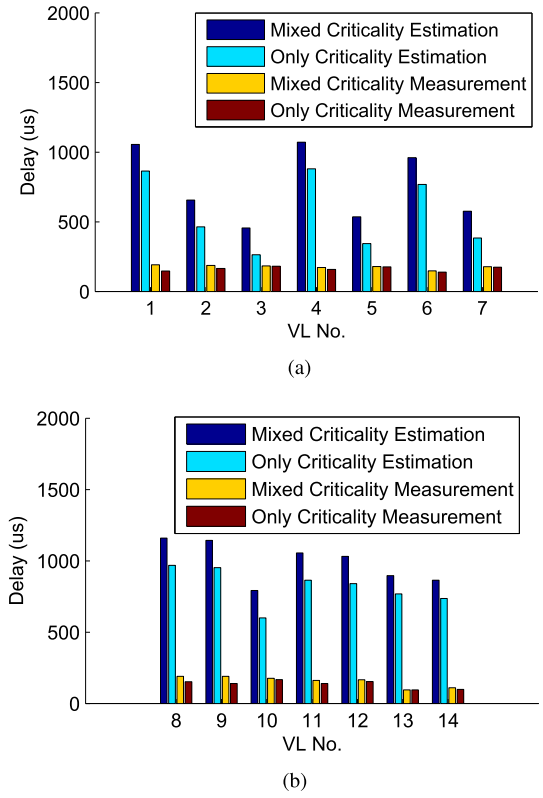


FIGURE 7. The delay comparisons of analyzed estimation and experimental measurement: (a) VL1-VL7; (b) VL8-VL14.

Fig. 7 shows the maximum delay values obtained by theoretical estimation and actual measurement for each VL, corresponding to mixed criticality architecture and only criticality architecture. From these results, it is obvious that the delay of each VL with mixed-criticality architecture is more than that with one criticality architecture. This is because of the blocking of non-critical traffics by the maximum frame, as discussed in Sec. IV. However, it is worth noting that the delay impacted by non-critical traffics is very small as can be seen from the experimental results. The extra delay in criticality architecture, as obtained by theoretical estimation, is $500 \times 8/100 = 40$us, whereas that in mixed

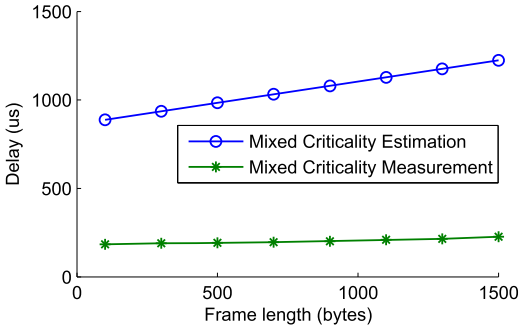


FIGURE 8. The delay comparison of analyzed estimation and experimental measurement with different maximum frame length of non-critical traffics.

criticality architecture and one criticality architecture, as obtained by actual measurement, is even smaller. This indicates that the worst-case situation is not frequently triggered in the actual network. To verify the extra delay analysis presented in Sec. IV, simulations of analyzed estimation and experimental measurement are conducted for delay comparison with different maximum frame lengths of non-critical traffics. In these simulations, the maximum frame length of non-critical traffics varies from 100 Bytes to 1500 Bytes. The simulation results presented in Fig. 8 show that the extra delay increases as the maximum frame length of non-critical traffics increases. From these results, we also see that the actual delay is much smaller than that obtained by theoretical estimation [3].

VI. CONCLUSION

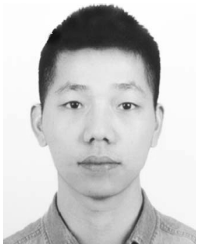
This study undertakes the worst-case delay analysis for mixed-criticality traffics in AFDX network. The proposed approach, which is based on compositional scheduling, suggests a method for integrating critical and non-critical traffics. In the process of integration, either the original network bandwidth is maintained or it is better utilized for non-critical traffics with limited impact on the performance of critical traffics. In this compositional scheduler, BAG-based scheduling is used to handle critical traffics with high priority, and Round Robin scheduler to handle non-critical traffics with low priority. The End-to-End worst-case delay estimates are derived by adopting the approach of Network Calculus. Finally, the proposed mixed-criticality architecture is verified and discussed through Truetime-based simulation in AFDX network.

REFERENCES

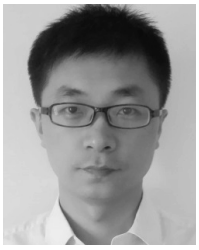
[1] J. Sifakis, "A vision for computer science—The system perspective," *Central Eur. J. Comput. Sci.*, vol. 1, no. 1, pp. 108–116, Mar. 2011.
[2] *Aircraft Data Network Part 7: Avionics Full-Duplex Switched Ethernet Network—ARINC Specification 664 P7-1*, ARINC, Annapolis, MD, USA, 2009.
[3] J. Yao and G. Zhu, "System-level scheduling of mixed-criticality traffics in avionics networks," in *Proc. Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2013, pp. 440–441.
[4] F. Ridouard, J.-L. Scharbarg, and C. Fraboul, "Stochastic upper bounds for heterogeneous flows using a static priority queueing on an AFDX network," in *Proc. 13th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Hamburg, Germany, Sep. 2008, pp. 1–11.

[5] I. Moir and A. Seabridge, *Aircraft Systems: Mechanical, Electrical and Avionics Subsystems Integration* (AIAA Education Series). Wiley, 2001.
[6] M. Tawk, G. Zhu, X. Liu, J. Li, Y. Savaria, and F. Hu, "Optimal scheduling and delay analysis for AFDX end-systems," SAE Tech. Paper 2011-01-2751, Oct. 2011.
[7] J. J. Gutiérrez, J. C. Palencia, and M. G. Harbour, "Holistic schedulability analysis for multipacket messages in AFDX networks," *Real-Time Syst.*, vol. 50, no. 2, pp. 230–269, Mar. 2014.
[8] S. Gurjar and B. Lakshmi, "Optimal scheduling policy for jitter control in AFDX end-system," in *Proc. Recent Adv. Innov. Eng. (ICRAIE)*, May 2014, pp. 1–4.
[9] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul, "Methods for bounding end-to-end delays on an AFDX network," in *Proc. 18th Euromicro Conf. Real-Time Syst.*, 2006, pp. 202–211.
[10] N. Rejeb, I. Mhadhb, A. K. Ben Sale, and S. Ben Saoud, "AFDX-CAN architecture for avionics applications," *Int. J. Comput. Sci., Commun. Inf. Technol.*, vol. 1, pp. 7–13, 2014. [Online]. Available: http://ipco-co.com/papers-CSCIT/CEIT_13_ID_164.pdf
[11] M. Fidler, "An end-to-end probabilistic network calculus with moment generating functions," in *Proc. 14th IEEE Int. Workshop Quality Service (IWQoS)*, Jun. 2006, pp. 261–270.
[12] J.-L. Scharbarg, F. Ridouard, and C. Fraboul, "A probabilistic analysis of end-to-end delays on an AFDX avionic network," *IEEE Trans. Ind. Informat.*, vol. 5, no. 1, pp. 38–49, Feb. 2009.
[13] G. Kemayo, N. Benammar, F. Ridouard, H. Bauer, and P. Richard, "Improving AFDX end-to-end delays analysis," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–8.
[14] Z. Wu, N. Huang, R. Li, and Y. Zhang, "A delay reliability estimation method for avionics full duplex switched Ethernet based on stochastic network calculus," *Eksploatacja Niezawodnosc*, vol. 17, no. 2, pp. 288–296, 2015.
[15] S. Bondorf and J. Schmitt, "Calculating accurate end-to-end delay bounds—You better know your cross-traffic," in *Proc. ValueTools*, 2015, pp. 17–24.
[16] I. Shin and I. Lee, "Compositional real-time scheduling framework," in *Proc. 25th IEEE Int. Real-Time Syst. Symp.*, Dec. 2004, pp. 57–67.
[17] I. Shin and I. Lee, "Compositional real-time scheduling framework with periodic model," *ACM Trans. Embedded Comput. Syst.*, vol. 7, no. 3, pp. 30:1–30:39, Apr. 2008.
[18] A. Easwaran, I. Lee, O. Sokolsky, and S. Vestal, "A compositional scheduling framework for digital avionics systems," in *Proc. 15th IEEE Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2009, pp. 371–380.
[19] Y. Bernet *et al.*, "A framework for integrated services operation over DiffServ networks," RFC Editor, Netw. Working Group, Tech. Rep. RFC 2998, 2000.
[20] J. Joung, "Feasibility of supporting real-time traffic in DiffServ architecture," in *Wired/Wireless Internet Communications* (Lecture Notes in Computer Science), vol. 4517. Heidelberg, Germany: Springer, 2007, pp. 189–200.
[21] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, 1973.
[22] T.-W. Kuo and A. K. Mok, "Load adjustment in adaptive real-time systems," in *Proc. 12th Real-Time Syst. Symp.*, Dec. 1991, pp. 160–170.
[23] E. Bini, G. C. Buttazzo, and G. M. Buttazzo, "Rate monotonic analysis: The hyperbolic bound," *IEEE Trans. Comput.*, vol. 52, no. 7, pp. 933–942, Jul. 2003.
[24] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Berlin, Germany: Springer-Verlag, 2001.

JIANGUO YAO received the B.E., M.E., and Ph.D. degrees from the Northwestern Polytechnical University (NPU), Xian, China, in 2000, 2007, and 2010, respectively. He was a Joint Post-Doctoral Fellow with the Ecole Polytechnique de Montreal and the McGill University in 2011. He was with Shanghai Jiao Tong University as an Associate Professor. His research interests are feedback control applications, real-time systems, cloud computing, and cyber-physical systems.



JIAHONG WU received the B.E degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2015. He is currently the master's degree with the Shanghai Jiao Tong University, Shanghai, China under the supervision of Prof. Y. Jianguo. His research interests are real-time systems, communication networks, and cyber-physical systems.



QINGCHUN LIU received the M.A degree in electrical engineering from the Nanjing University of Aeronautics and Astronautics, Jiangsu, China. He is currently a Senior Systems Engineer with the Science and Technology on Avionics Integration Laboratory. His recent research interests include system architecture, applications of model-based systems engineering.

ZHIYONG XIONG received the M.A. degree in computer and electrical engineering from Northwestern Polytechnical University, China. He is currently an Assistant Chief Engineer with the Science and Technology on Avionics Integration Laboratory. His recent research interests include architectural design methods, technology planning.

GUCHUAN ZHU received the M.S. degree in electrical engineering from the Beijing Institute of Aeronautics and Astronautics, Beijing, China, the Ph.D. degree in mathematics and control from the Ecole des Mines de Paris, Paris, France, and the Graduate Diploma degree in computer science from Concordia University, Montreal, QC, Canada, in 1982, 1992, and 1999, respectively. He is currently an Associate Professor with Ecole Polytechnique de Montreal, Montreal, QC, Canada. His research interests include distributed parameter system control, nonlinear and robust control, and optimization with their applications to microsystems, communication networks, and the smart grid.