Trans. Japan Soc. Aero. Space Sci.
Vol. 64, No. 1, pp. 1–12, 2021
DOI: 10.2322/tjsass.64.1

Performance Evaluation on Packet Transmission for Distributed Real-time Avionics Networks Using Forward End-to-End Delay Analysis*

Qingfei XU[1),2)†] and Xinyu YANG[1)]

[1)]*School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2)]*The fiftieth Research Institute of China Electronic Technology Group Corporation, Shanghai 200331, China*

With growing avionics applications, the transmission of avionics data flows has been increasing in real-time avionics networks of aircraft. An Avionics Full DupleX switched Ethernet (AFDX), standardized as ARINC 664, is chosen as the backbone network for distributed real-time avionics systems as it offers high throughput and does not require global clock synchronization. Estimating the end-to-end transmission delay to validate the network performance is essential for both certification and industrial research. Because of the various waiting times caused by the backlog (i.e., the pending packets in the output port of the visited switch), it is necessary and reasonable to compute the worst-case end-to-end transmission delay to validate network performance. Several approaches have been designed to compute the upper boundaries of end-to-end transmission delays, such as the Network Calculus approach and the Trajectory approach. In this paper, we focus on a new approach, Forward end-to-end delay Analysis (FA). This approach iteratively estimates the maximum backlog (i.e., number of pending packets) in each switch visited along the transmission path, so that the worst-case end-to-end transmission delay can be computed and the network performance evaluated. We also present the termination condition for this iterative estimation. The experiments demonstrate that this approach achieves a more accurate evaluation of transmission performance than the Network Calculus approach. A comparison with the exact upper boundaries obtained using the Model Checking approach shows the pessimism (i.e., overestimation) in FA. This paper analyses the reasons for that pessimism, and proposes future research.

Key Words: Performance Evaluation, Worst-case Delay, Backlog, Serialization Effect

Nomenclature

$F_{\max}$: maximum packet length in an AFDX network
C_i: maximum transmission time of a packet generated by flow v_i
T_i: minimum time interval of two consecutive packets generated by flow v_i
$\mathcal{P}_i$: transmission path of flow v_i
$|\mathcal{P}_i|$: the count of nodes in path $\mathcal{P}_i$
L: switching fabric delay, upper-bounded to 16 µs
m_i: a packet generated by flow v_i
$Smax_i^h$: maximum end-to-end transmission delay of packet m_i from its source node to node h
$Smin_i^h$: minimum end-to-end transmission delay of packet m_i from its source node to node h
m_j: the last packet of flow v_j arriving at node h at the end of time interval $[a, b]$ suffering the minimum delay ($Smin_j^h$)
m'_j: the last packet arriving at node h suffering the maximum delay ($Smax_j^h$)
G: maximum servicing rate of a node in an AFDX network, usually 100 Mb/s
R_i: worst-case end-to-end transmission delay of a packet generated by flow v_i
R_i^h: maximum transversal delay from the generation time of packet m_i at its source node to its departure time at node h
$\mathcal{S}$: node set of an AFDX network
θ: the time that packet m'_j arrives at node h
ε: the time interval from the arrival of packet m_j of flow v_j to the end of time interval $[a, b]$
Δ_j^h: the time difference from time θ to the arrival time of packet m_j
t_{ref}: generation time of packet m'_j
Γ: flow set transmitted over an AFDX network
J_j^h: worst-case arrival jitter of flow v_j at node h
$\mathcal{BP}^h$: the upper bound of the test time interval for the periodic part at node h
$\mathcal{BS}^h$: the upper bound of the test time interval for the simultaneous part at node h
$\mathcal{B}^h$: the upper bound of the test time interval t at node h
$|h|^i$: the number of nodes crossed by packet m_i from $first_i$ to h
α_i^h: the time that packet i arrives at node h
$RBF_j^h(t)$: total transmission time of packets generated by flow v_j arriving at node h during time interval t
$W(t)$: total transmission time of packets generated by all of the flows passing through node h during time interval t
Γ_h: flow set transmitted through node h
$first_i$: source node of flow v_i along path $\mathcal{P}_i$
$last_i$: destination node of flow v_i along path $\mathcal{P}_i$
$Bklg_i^h$: transmission time of the maximum pending packets at node h when packet m_i arrives

© 2021 The Japan Society for Aeronautical and Space Sciences
*Received 24 May 2019; final revision received 4 April 2020; accepted for publication 15 June 2020.
†Corresponding author, qingfeixu@foxmail.com

1. Introduction

The transmission performance[1,2] of real-time avionics networks in aircraft is quite important for safety-critical systems such as flight control and dynamics.

Current aircraft are composed of different systems such those for dynamics control, fuel control, power management, environmental control, lighting and power distribution. Each system is composed of a communication network and exchanges data packets with the networks of other systems. Different networks use different communication buses. For example, the dynamics control and fuel control systems in a Boeing 787 or Airbus 380 use Avionics Full DupleX switched Ethernet (AFDX), the power management system in a Boeing 787 and the environmental control system in an Airbus 380 use Time-Triggered Protocol (TTP), and the lighting and power distribution systems in a Boeing 787 or Airbus 380 use the Controller Area Network (CAN).[3] Different buses require different types of cables and connectors that increase aircraft weight. Furthermore, data packets in different systems have different transmission formats (i.e., packets transmitting from one network to another require format conversion). All of these things influence packets transmission performance and aircraft reliability.

In the future, a common network will cover all the systems of an aircraft. All data packets from different systems will have the same data format and be transmitted over a shared communication network.[4] There will be only one type of connector and cable, so that data format conversion is eliminated. Furthermore, the weight of aircraft can be reduced as the number of cables and connectors will be reduced. With the growing need for functionality and communication, the exchange of data packets is increasing significantly in distributed real-time avionics systems. An immediate response to pilot operation is of significance because it relates directly to aircraft reliability and flight control performance. A deterministic upper boundary of the end-to-end transmission delay of each packet should be guaranteed. It is closely related to the reliability of aircraft and helpful for designing the communication networks in aircraft.

AFDX standardized as ARINC 664[5,6] is becoming the backbone network owing to its high throughput and reliability. To guarantee transmission performance it is mapping multiple predefined data flows over different physical links and is widely used for distributed real-time avionics networks.

An AFDX network is usually a distributed architecture without global clock synchronization. It is composed of end systems (ES), switches, and physical links. The network ingress/egress points are called end systems.[7] They are interconnected with switches through the physical links. The end systems in AFDX networks are packet transmitters and receivers. Packet transmission over the distributed real-time avionics networks should be completed within a limited time interval.[8,9] The end-to-end transmission delay of a packet corresponds to the duration from its generation time at the transmitting-end system to its arrival time at the receiving-end system. The transmission delay includes both the transmission time on the physical links and the waiting time in the output port of each visited switch. The former is constant: it is determined by the maximum packet length (less or equal to 1,518 bytes, the maximum packet length of the Ethernet) and the transmitting rate of the physical links (typically 100 Mb/s). The latter is quite various: it is dependent on the backlog (i.e., number of pending packets) in the output port of each visited switch. Because of the variety of waiting times, aircraft manufacturers usually estimate the maximum backlog at each visited switch along the transmission path, and an upper boundary of the end-to-end transmission delays is computed. This enables the transmission performance of an avionics network to be evaluated.

Forward end-to-end delay Analysis (FA)[10,11] is currently a popular approach that is used to evaluate the performance of real-time avionics networks by computing an upper boundary for end-to-end transmission delays. It iteratively analyzes the maximum backlog[12–14] in each visited switch along the packet transmission path. It computes the worst-case waiting time suffered by the packets of different flows over the network, so that the worst-case end-to-end transmission delays can be computed and the network performance evaluated.

The rest of this paper is organized as follows. The related work is given in Section 2. Section 3 presents the content of AFDX. FA is explained in Section 4: it includes constructing the worst-case scenario for the maximum backlog, the iterative computation of the delay upper boundary, and the termination condition of this iteration. Section 5 includes the comparative experiments performed for both a sample configuration and an industrial network. A comparison with other approaches is done. Section 6 provides a conclusion and discusses future research.

2. Related Work

Considerable research has been devoted to the performance evaluation of distributed real-time avionics networks by computing end-to-end transmission delays.

Simulation methods[15–17] can compute the distribution of end-to-end transmission delays for a given flow. However it can only estimate an average transmission delay. It cannot compute an upper boundary for end-to-end transmission delays as it is missing some rare events, such as the worst-case scenario. Moreover, the precision depends on the design of the model used in the simulation.

The Model Checking approach[18,19] can compute the end-to-end transmission delays of a given flow for all of the possible scenarios through an exhaustive search. However, this approach cannot manage real avionics networks with large configurations due to the combinatorial explosion problem. Nevertheless, the obtained result is a good reference for the exact worst-case end-to-end delays for small illustrative configurations.

The Network Calculus approach[20–22] is proposed to compute the worst-case end-to-end transmission delays for per-

©2021 JSASS

formance evaluation. This approach models packet flows as upper-traffic envelopes and switches as lower-service curves. Traffic envelopes of different packet flows sharing the same physical link can be summed as an aggregated envelope for this common physical link. The upper boundary of the transmission delay for a packet flow passing through a given switch is obtained by measuring the maximum horizontal distance between the traffic envelope of the packet flow and the service curve of the given switch.[23,24] The difference between the best and worst transmission delays at a switch is used as the input jitter at the next switch along the transmission path. The calculation propagates from the ingress point to the egress point, and the worst-case end-to-end transmission delay can be computed; thus, the network transmission performance can be evaluated. The result is pessimistic because of the cumulative input jitters.

The Trajectory approach[25–27] is another approach used evaluating the performance of network transmission. It is based on the concept of busy periods.[28,29] This approach iteratively analyses each node (i.e., output port of a visited switch or emitting end system) along the transmission path. It estimates the worst-case delays caused by the competing flows at each node so that there is no combinatorial explosion problem, and an upper boundary of the end-to-end transmission delays can be computed. This approach overcomes the "sum of local worst-cases," which is the basis of the Network Calculus approach. It can compute a more accurate upper boundary of the end-to-end transmission delays than the Network Calculus approach for most scenarios. Recent studies have found that the Trajectory approach might provide an optimistic (underestimated) upper boundary in some corner cases, but it is not suitable for evaluating the performance of distributed real-time avionics networks.[30,31]

The usage limitations in the above approaches require researchers to find a new method to evaluate the transmission performance of real-time avionics networks. This paper presents the FA approach, which can compute worst-case end-to-end transmission delays. It can scale up to real large avionics networks without experiencing the combinational explosion problem. It can also be applied to networks having nodes with heavy traffic loads.

3. AFDX Network

AFDX[5] is a packet-switched network based on Ethernet taking into account avionics constraints. It is an asynchronous and distributed real-time network without global clock synchronization. Full duplex connection ensures that there is no packet transmission collision at physical links. Each physical link has a maximum transmission speed (typically 100 Mb/s). Each switch has a FIFO buffer in every output port, but no buffers in input ports. The delay incurred by the switching fabric is upper-bounded to a constant value. An illustrative example of an AFDX network is depicted in Fig. 1, which includes 10 end systems (e_1–e_{10}) and five switches (S_1–S_5).

Packet flows are transmitted through predefined logical paths called "virtual links" (VLs).[32,33] A virtual link defines a logical and unidirectional connection from one transmitting-end system to one or more receiving-end systems. As shown in the example in Fig. 1, v_{10} is a unicast VL with path e_3–S_3–S_4–e_8, while v_6 is a multicast VL with paths e_1–S_1–S_2–e_7 and e_1–S_1–S_4–e_8. A virtual link is determined by the maximum packet length ($F_{\max}$) and minimum time interval between the generation of two consecutive packets of the given VL. This minimum time interval is called "Bandwidth Allocation Gap (BAG)."

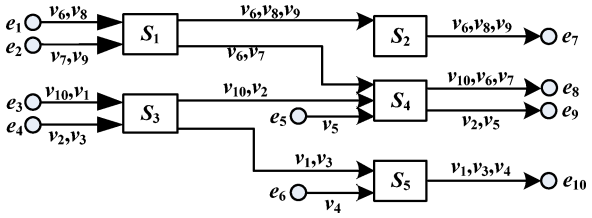


Fig. 1. Illustrative AFDX network configuration.

The deterministic behavior of the AFDX is ensured by traffic-shaping and the policy unit in each end system or switch. Typically, an industrial AFDX network includes more than 100 end systems and two redundant AFDX sub-networks, each composed of at least 10 switches. Nearly 1,000 VLs are transmitted on each sub-network, corresponding to more than 6,000 paths due to the multicast characteristic of VLs.[34]

4. FA Approach

4.1. Modeling of network and traffic

A real-time avionics network is composed of a set of end systems and switches that are interconnected through physical links. Each multiplexing point (i.e., output port of an end system or a switch) can be modeled as a node. A node can be seen as a FIFO buffer that is working at the maximum servicing rate $G = 100\,\text{Mb/s}$. All of these nodes constitute a node set, denoted as $\mathcal{S} = \{N_1, N_2, \ldots, N_p\}$. The switching fabric delay is denoted as L, and it is upper-bounded to $16\,\mu\text{s}$.[10,35]

Packet flows transmitted through the switches are composed of a flow set, denoted as $\Gamma = \{v_1, v_2, \ldots, v_n\}$. The subset Γ_h includes all of the flows passing through node h. The transmission path $\mathcal{P}_i$ of flow v_i is statically defined as an ordered node list from the source node ($first_i$) to the destination node ($last_i$). Here, the source node ($first_i$) is the transmitting-end system where flow v_i is generated. The destination node ($last_i$) is not the receiving-end system, but the output port of the last visited switch along the transmission path $\mathcal{P}_i$.

Generally, a flow v_i can be characterized as follows:

- C_i, the maximum transmission time of a packet generated by flow v_i. It is equal to the maximum packet length $F_{\max}$ divided by the switch servicing rate G:

$$C_i = \frac{F_{\max}}{G}$$

- T_i, the minimum time interval between two consecutive packets generated by flow v_i. It corresponds to the BAG of a given VL in AFDX.
- $\mathcal{P}_i$, the transmission path of flow v_i. An ordered list of the nodes crossed, from the source node ($first_i$) to the destination node ($last_i$).
- L, the switching fabric delay. It is upper-bounded to 16 µs.

4.2. Principles

Provided that m_i is a packet generated by flow v_i, in order to compute the worst-case end-to-end transmission delay, FA iteratively analyses the maximum backlog at each node in path $\mathcal{P}_i$. It estimates the maximum transversal delay from the generation time at its source node to the departure time from each visited node h of path $\mathcal{P}_i$. We denote this delay as R_i^h. The estimation propagates until the destination node ($last_i$), so that the worst-case end-to-end transmission delay can be computed.[10)]

DEFINITION 1. For any node h of path $\mathcal{P}_i$, the term $Smax_i^h$ is the maximum transmission delay for a packet of flow v_i from its generation time at the source node ($first_i$) to its arrive time at node h.

DEFINITION 2. For any node h of path $\mathcal{P}_i$, the term $Smin_i^h$ is the minimum transmission delay for a packet of flow v_i from its generation time at the source node ($first_i$) to its arrival time at node h.

DEFINITION 3. Backlog is defined as the total transmission time of all of the pending packets in an output buffer at a given time. We define $Bklg_i^h$ as the total transmission time of the maximum pending packets in the output buffer of node h when packet m_i arrives.

Suppose node h belongs to path $\mathcal{P}_i$, and node $h+1$ follows node h along path $\mathcal{P}_i$. As each node follows the FIFO scheduling policy without considering priority, packet m_i cannot be delayed by the packets arriving after itself, $Smax_i^h$ can be estimated iteratively as depicted in Fig. 2.[10,11)]

We define the generation time of packet m_i at the source node ($first_i$) as the origin time, so that we have $Smax_i^{first_i} = Smin_i^{first_i} = 0$. $Smax_i^{h+1}$ can be obtained as shown below if the value of $Smax_i^h$ is known[10)]:

$$Smax_i^{h+1} = Smax_i^h + Bklg_i^h + C_i + L \tag{1}$$

This computation propagates until the destination node ($last_i$), and the worst-case end-to-end delay R_i of packet m_i is equal to the maximum transversal delay $R_i^{last_i}$. It can be expressed as Eq. (2).

$$R_i = R_i^{last_i} = Smax_i^{last_i} + Bklg_i^{last_i} + C_i \tag{2}$$

To compute an upper boundary of the end-to-end transmission delay of flow v_i, we need to determine the maximum backlog $Bklg_i^h$ at each crossed node h along path $\mathcal{P}_i$.

For $Smin_i^h$, it occurs if there are no pending packets in the output buffer when packet m_i arrives; thus, packet m_i can be immediately forwarded once it arrives. When there is no backlog at each visited node, $Smin_i^h$ can be obtained as Eq. (3); where $|h|^i$ denotes the number of nodes crossed by packet m_i from $first_i$ to h.

$$\begin{cases} Smin_i^{h+1} = Smin_i^h + C_i + L \\ Smin_i^h = (|h|^i - 1) \times (C_i + L) \end{cases} \tag{3}$$

The FA approach evaluates the maximum backlog $Bklg_i^h$ at each node h along path $\mathcal{P}_i$ using $Smin_i^h$ and $Smax_i^h$, and the worst-case end-to-end transmission delay R_i can be computed.[35)]

4.3. Worst-case scenario with maximum backlog

This section presents the worst-case scenario, which can construct the maximum backlog at node h. We focus on packet m_i generated by flow v_i. The maximum backlog $Bklg_i^h$ that can delay packet m_i at node h is bounded by the time window from its earliest arrival time ($Smin_i^h$) to its latest arrival time ($Smax_i^h$).

THEOREM 1.[10)] The maximum backlog generated by an interfering flow v_j at node h during a time interval $[a, b]$ occurs in the scenario described below:

(i) Packets of flow v_j should be generated at the maximum rate from the source node ($first_j$).

(ii) All of the packets of flow v_j arriving at node h in the time interval $[a, b]$ constitute a packet sequence ω. Suppose that m_j is the last packet of ω, it has to arrive at node h strictly at the end of this time interval (i.e., time instant b).

(iii) The last packet m_j in the packet sequence ω arrives at node h suffering the minimum delay ($Smin_j^h$). In the packet sequence ω, there is another packet m'_j before m_j; it is the last packet that arrives at node h earlier than m_j, suffering the maximum delay ($Smax_j^h$).

(iv) All of the packets before m'_j in the packet sequence ω arrive at node h suffering the maximum delay ($Smax_j^h$).

(v) Suppose that there are k packets between m'_j and m_j. All of these k packets arrive at node h simultaneously with packet m_j.

The corresponding worst-case scenario is depicted as Fig. 3.

Proof: To construct the maximum backlog, the interfering flow v_j should generate the maximum competing traffic. To achieve this, packets of flow v_j should be generated at the source node ($first_j$) as quickly as possible; hence, flow v_j should generate packets at the minimum time interval T_j, which proves item (i).[35)]

Suppose packet m_j of flow v_j arrives at node $h\varepsilon$ earlier

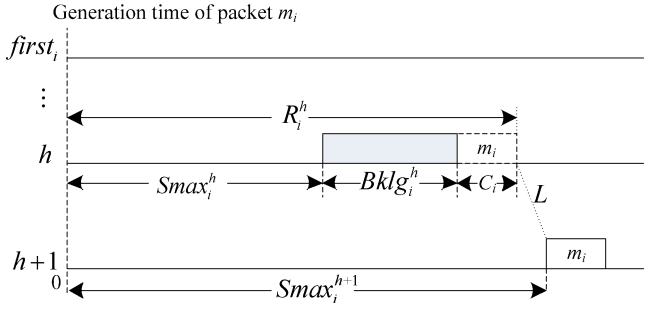


Fig. 2. Iterative estimation of $Smax_i^h$ and R_i^h.

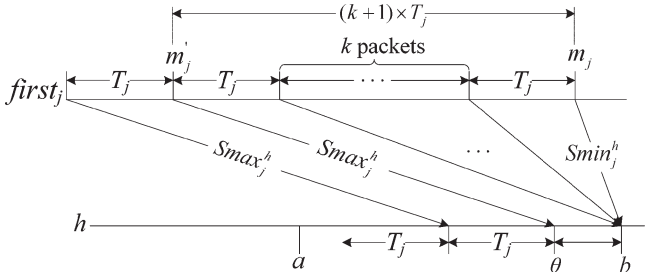


Fig. 3. Worst-case scenario that packet m_j arrives at node h simultaneously with other previous k packets.

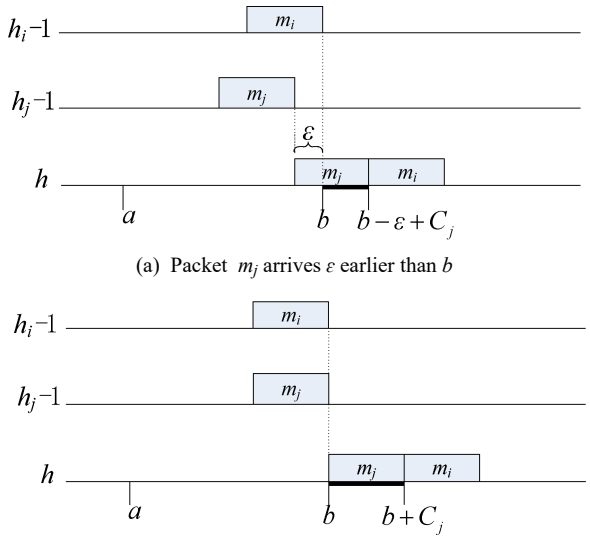


Fig. 4. Packet m_j delays packet m_i.

than b. When packet m_i arrives at node h strictly at time b as depicted in Fig. 4(a), it can be forwarded at time $b + C_j - \varepsilon$. This means packet m_i suffers $C_j - \varepsilon$ delay, which is marked as a black bar. If packet m_j of flow v_j arrives at node h strictly at time b as depicted in Fig. 4(b), when packet m_i arrives at node h strictly at time b, it can be forwarded at time $b + C_j$. This means packet m_i suffers C_j delay (also marked as a black bar), which is longer than $C_j - \varepsilon$.

No matter how much earlier than b packet m_j arrives at node h, there is always a worse scenario that packet m_j arrives at node h strictly at time b, and this scenario can cause a longer delay to packet m_i. Item (ii) is proved.[10)]

If the last packet m_j arrives at node h suffering a delay $d\,(Smin_i^h < d < Smax_i^h)$, the time difference between the arrivals of the last two packets (m_j and its previous packet) in the packet sequence ω is larger than when packet m_j arrives, suffering the minimum delay $Smin_i^h$, so that the backlog is not maximized. Some packets in the packet sequence ω will not arrive simultaneously with packet m_j. To ensure the maximum backlog, they should suffer the maximum delay to arrive at node h. Item (iii) is proved.[35)]

Since packet m_j' arrives at node h in the time interval $[a, b]$ and suffers the maximum delay $Smax_j^h$, to ensure more packets can arrive at node h during this time interval, all of the packets before m_j' have to suffer the maximum delay $Smax_j^h$. Item (iv) is proved.

For item (v), the k packets arrive at node h no earlier than m_j', and based on item (ii), they should arrive at node h simultaneously with m_j to ensure the maximum delay to packet m_i. The last item is proved.[10)]

Generally, we can analyze the packet sequence ω from back to front.

- The last packet m_j arrives at node h strictly at the end of the time interval $[a, b]$, suffering the minimum delay $Smin_j^h$.
- There are k packets before packet m_j, and these k packets arrive at node h simultaneously with packet m_j, suffering different delays between ($Smin_j^h$, $Smax_j^h$).
- There is a packet m_j' arriving at node h earlier than the end of the time interval $[a, b]$, suffering the maximum delay $Smax_j^h$.
- The other packets before m_j' in the packet sequence ω arrive at node h, suffering the maximum delay $Smax_j^h$.

THEOREM 2.[10)] The difference in the arrival times at node h between packets m_j' and m_j is not longer than the minimum packet generation interval T_j.

Proof: Suppose that packet m_j' arrives at node h at time θ, as depicted in Fig. 3. From items (ii) and (iii), we have $\theta < b$. Packet m_j' suffers the maximum delay $Smax_j^h$. The next packet T_j is generated after m_j', and it won't suffer a delay longer than $Smax_j^h$. Therefore, the difference in arrival time between those two packets (i.e., m_j' and the following one) is not longer than T_j. The following packet arrives at node h simultaneously with packet m_j; thus, the arrival time difference between packets m_j' and m_j is not longer than T_j. Theorem 2 is proved.

THEOREM 3.[10)] All of the packets before m_j' arrive at node h periodically, with time interval T_j.

Proof: Item (iv) demonstrates that all of the packets before m_j' arrive at node h, suffering the maximum delay $Smax_j^h$. Since packet m_j' also suffers the maximum delay, and the generation interval for each packet is T_j, packets arrive at node h one after another at time interval T_j. Theorem 3 is proved.

4.4. Request boundary function

The maximum backlog generated by flow v_j at node h during a time interval $[a, b]$ was presented in Theorem 1. In order to determine this backlog, we define the Request Boundary Function (RBF) and a cumulative function $W(t)$ in this section.

The RBF $RBF_j^h(t)$ defines the total transmission time of packets generated by flow v_j arriving at node h during a time interval with the length of $t\,(t = b - a)$. The cumulative function $W^h(t)$ defines the total transmission time of the packets generated by all of the flows passing through node h during a time interval of length t. We have[10)]:

$$W^h(t) = \sum_{v_j \in \Gamma_h} RBF_j^h(t) \tag{4}$$

The computation of $RBF_j^h(t)$ has to be consistent with the scenario built as Theorem 1. As depicted in Fig. 3, the back-

log at node h generated by flow v_j is comprised of two parts:

(a) The periodic part ending with packet m'_j (i.e., m'_j is included). Packets in this part arrive at node h one after another with a time interval T_j.

(b) The simultaneous part ending with packet m_j (i.e., m_j is included). Packets in this part arrive at node h simultaneously with m_j.

LEMMA 1.[35] The maximum backlog generated by flow v_j at node h is achieved when packet m'_j arrives at node h at time θ. Suppose θ is Δ_j^h earlier than b, then we have:

$$\Delta_j^h = (k+1) \times T_j - J_j^h \tag{5}$$

where $J_j^h = Smax_j^h - Smin_j^h$ is the worst-case arrival jitter of flow v_j, and k is the number of packets arriving at node h simultaneously with packet m_j.

Proof: Consider the generation time of packet m'_j as the reference time t_{ref}. Its arrival time at node h can be expressed as:

$$\theta = t_{ref} + Smax_j^h \tag{6}$$

As there are $k+1$ time intervals T_j from m'_j to m_j, as depicted in Fig. 3, and packet m_j arrives at node h suffering the minimum delay $Smin_j^h$, the time instant b can be expressed as:

$$b = t_{ref} + (k+1) \times T_j + Smin_j^h \tag{7}$$

and Lemma 1 can be proved as below:

$$\begin{aligned} \Delta_j^h &= b - \theta \\ &= (k+1) \times T_j + Smin_j^h - Smax_j^h \\ &= (k+1) \times T_j - J_j^h \end{aligned} \tag{8}$$

LEMMA 2.[10] The obtained term Δ_j^h (c.f. Lemma 1) is always greater than zero: $\Delta_j^h > 0$.

Proof: If $\Delta_j^h = 0$, this means that packet m'_j arrives at node h simultaneously with m_j. This is contrary to item (iii) of Theorem 1, which demonstrates packet m'_j arrives earlier than m_j. If $\Delta_j^h < 0$, this means that packet m'_j arrives at node h later than m_j, as depicted in Fig. 5. Packet m_j is generated after m'_j, and following the FIFO scheduling policy, it definitely arrives at node h after m'_j. It is impossible that $\Delta_j^h < 0$.

LEMMA 3.[10] Let k be the number of packets arriving at node h simultaneously with packet m_j. In the scenario with the worst-case backlog, k satisfies:

$$kT_j \le J_j^h < (k+1)T_j \tag{9}$$

Proof: Lemma 2 has proved that $\Delta_j^h > 0$; therefore, from Eq. (8), we can get $(k+1) \times T_j - J_j^h > 0$, and $J_j^h < (k+1)T_j$, and the right side of Eq. (9) is proved. Theorem 2 demonstrates that $\Delta_j^h \le T_j$, which means that $(k+1) \times T_j - J_j^h \le T_j$ and $kT_j \le J_j^h$; thereby proving the left side of Eq. (9).

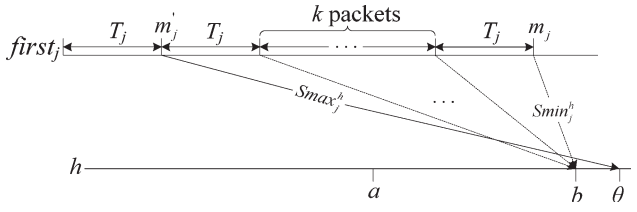


Fig. 5. Packet m'_j arrives later than m_j.

THEOREM 4.[35] RBF $RBF_j^h(t)$ for flow v_j passing through node h at a time interval of length t ($t > 0$) is:

$$RBF_j^h(t) = \left(1 + \left\lfloor \frac{t + J_j^h}{T_j} \right\rfloor\right) \cdot C_j \tag{10}$$

Proof: We separately prove Eq. (10) for two different parts: the periodic part ending with packet m'_j and the simultaneous part ending with packet m_j.

When length t is shorter than Δ_j^h, the maximum count of the pending packets is $k+1$ (i.e., packet m_j and the k packets arriving simultaneously with it, as depicted in Fig. 5). From Eq. (9), we have $kT_j \le J_j^h$, so that $kT_j \le t + J_j^h$. Since t is shorter than Δ_j^h, we have $t + J_j^h < J_j^h + \Delta_j^h$. From Eq. (8), we have $J_j^h + \Delta_j^h = (k+1) \times T_j$, so that $t + J_j^h < (k+1) \times T_j$. We have:

$$\begin{aligned} & kT_j \le t + J_j^h < (k+1)T_j \\ \Leftrightarrow\ & k \le \frac{t + J_j^h}{T_j} < (k+1) \\ \Leftrightarrow\ & \left\lfloor \frac{t + J_j^h}{T_j} \right\rfloor = k \end{aligned} \tag{11}$$

so that Eq. (10) can get the maximum backlog for any time interval t ($0 < t < \Delta_j^h$).

When the length t is Δ_j^h, the maximum count of the pending packets is $k+2$ (i.e., packet m_j and the k packets arriving simultaneously with it and packet m'_j, as depicted in Fig. 5). We have:

$$\begin{aligned} t + J_j^h &= \Delta_j^h + J_j^h \\ &= (k+1)T_j \\ \Leftrightarrow\ & \left\lfloor \frac{t + J_j^h}{T_j} \right\rfloor = k+1 \end{aligned} \tag{12}$$

Equation (10) can get the maximum backlog for time interval $t = \Delta_j^h$ and $t + J_j^h$ is a multiple of T_j.

When the length t is longer than Δ_j^h, one new packet can arrive every T_j because all of the packets before m'_j arrive at node h one after another every T_j.

The above analysis proves Theorem 4.

COROLLARY 1.[10,35] Equation (10), defined in Theorem 4, is identical at the source node to the total transmission time of the pending packets generated by flow v_j during a time interval of length t.

Proof: By the definition, at the source node, we have $Smin_j^{first_j} = Smax_j^{first_j} = 0$. Therefore, $J_j^{first_j} = 0$ and Eq. (10) can be expressed as:

$$RBF_j^h(t) = \left(1 + \left\lfloor \frac{t}{T_j} \right\rfloor\right) \cdot C_j \tag{13}$$

As illustrated by Corollary 1, at the source node, every time interval T_j, one new packet of flow v_j can be generated. There is only the periodic arriving part at the source node.

©2021 JSASS

With RBF $RBF_j^h(t)$ computed using Eq. (10), we can obtain the total transmission time $W^h(t)$ for the packets generated by all of the interfering flows during a time interval of length t. On one hand, the competing packets arrive forming the backlog at node h, and on the other hand, the arrived packets are being forwarded leaving node h. Therefore, the available time t to forward those packets should be subtracted. Furthermore, Fig. 2 and Eq. (1) demonstrate that packet m_i is included in the computation of $Smax_i^{h+1}$. Accordingly, the transmission time C_i should be subtracted during computation of the maximum backlog $Bklg_i^h$. We have:

$$Bklg_i^h = \max_{t\geq 0}(W^h(t) - C_i - t) \tag{14}$$

COROLLARY 2.[10] $Bklg_i^h$ computed using Eq. (14) is never negative.

Proof: When there is only one flow v_i without any interfering flows, for the minimum time interval $t = 0$, we obtain $W^h(0) - C_i - 0 = 0$. This is the minimum backlog. However, $Bklg_i^h$ corresponds to the maximum value of this computation, so it cannot be negative.

4.5. Stop condition

To compute the maximum backlog $Bklg_i^h$ encountered by flow v_i at node h, the time interval of any possible length t should be tested. It is necessary to determine a maximum length as the stop condition of this test to ensure the maximum backlog is included. We denote the maximum length of the time interval as $\mathcal{B}^h$. To determine the maximum length, we respectively take into account the periodic part and the simultaneous part in Section 4.4.

For the periodic part, we introduce the concept of busy period.[23,29] The backlog is included in a busy period. A busy period is a time interval between two consecutive idle time instants that are respectively the starting time and ending time of the busy period. An idle time instant means that all of the pending packets have been forwarded (i.e., there is no backlog at this idle time instant). Therefore, packets that arrive before an idle time instant should not be included in computation of the backlog for the following busy period. Consequently, computation of the backlog should end with the first following idle time instant (i.e., ending time of the busy period), as packets arriving after the idle time instant cannot generate any backlog.

LEMMA 4.[11] The length of the maximum busy period at node h generated by flow set Γ_h is bounded to the first positive value satisfying the following equation if

$$\sum\nolimits_{j\in\Gamma_h} \frac{C_j}{T_j} \leq 1.$$

$$\begin{cases} \mathcal{BP}^h(0) = \sum\limits_{j\in\Gamma_h} C_j \\ \mathcal{BP}^h(k+1) = \sum\limits_{j\in\Gamma_h} \left\lceil \dfrac{\mathcal{BP}^h(k)}{T_j} \right\rceil \cdot C_j \end{cases} \tag{15}$$

This computation ends with the first positive value satisfying $\mathcal{BP}^h(k+1) = \mathcal{BP}^h(k)$.

The longest busy period represents the maximum time interval between two consecutive idle time instants. Therefore, for the periodic part of a flow, if the length of a time interval is larger or equal to the longest busy period $\mathcal{BP}^h$, computation of the backlog can be stopped as an idle time exists.[10]

For the simultaneous part, as depicted in Lemma 1, Δ_j^h represents its length. Generally, $\max_{j\in\Gamma_h}(\Delta_j^h)$ can include all of the simultaneous parts of the flows passing through node h. Δ_j^h can be determined by Lemma 5 when the maximum jitter J_j^h is known.

LEMMA 5.[35] For any flow v_j passing through node h, we have:

$$\Delta_j^h = \left(1 + \left\lfloor \frac{J_j^h}{T_j} \right\rfloor\right) \cdot T_j - J_j^h \tag{16}$$

Proof: From Eq. (8), we obtain $\Delta_j^h = (k+1) \times T_j - J_j^h$; where k is the number of the packets arriving simultaneously with packet m_j and J_j^h is the maximum arrival jitter ($J_j^h = Smax_j^h - Smin_j^h$). From Eq. (9), we have:

$$\begin{aligned} & kT_j \leq J_j^h < (k+1)T_j \\ \Leftrightarrow\ & k \leq \frac{J_j^h}{T_j} < (k+1) \\ \Leftrightarrow\ & \left\lfloor \frac{J_j^h}{T_j} \right\rfloor = k \end{aligned} \tag{17}$$

Replacing k with $\lfloor J_j^h/T_j \rfloor$ in Eq. (8) proves Lemma 5.

The maximum length of the time interval containing all simultaneous parts can be limited to $\mathcal{BS}^h$:

$$\mathcal{BS}^h = \max_{v_j\in\Gamma_h} \left\{ \left(1 + \left\lfloor \frac{J_j^h}{T_j} \right\rfloor\right) \cdot T_j - J_j^h \right\} \tag{18}$$

The two lemmas above determine the maximum length of the time interval for computing the maximum backlog respectively considering the periodic and simultaneous parts.

THEOREM 5.[10] Suppose a set of flows Γ_h passing through node h, if

$$\sum\nolimits_{j\in\Gamma_h} \frac{C_j}{T_j} \leq 1$$

and node h follows the FIFO servicing policy, the maximum length of the test time interval can be expressed as:

$$\mathcal{B}^h = \mathcal{BP}^h + \mathcal{BS}^h \tag{19}$$

Proof: Lemmas 4 and 5 prove this theorem.

4.6. Algorithm

For each flow $v_i \in \Gamma$ and each node h of path $\mathcal{P}_i$, an upper boundary of the worst-case transversal delay R_i^h can be obtained by

$$R_i^h = Smax_i^h + Bklg_i^h + C_i \tag{20}$$

The maximum backlog $Bklg_i^h$ can be computed using Eq. (21)

$$Bklg_i^h = \max_{\mathcal{B}^h\geq t\geq 0} (W^h(t) - C_i - t) \tag{21}$$

©2021 JSASS

Algorithm 1. FA approach for computing worst-case end-to-end transmission delay.

FA Approach

Input: Network configuration defined by S and Γ.
Switching fabric delay $L = 16$ µs.
For each flow $v_i \in \Gamma$:
$Fmax_i$, T_i, $\mathcal{P}_i$
begin
  for each *flow* $v_i \in \Gamma$ **do**
    $Smin_i^{first_i} = 0$
    $Smax_i^{first_i} = 0$
    $C_i = Fmax_i / G$
  end
  for each node $h \in \mathcal{P}_i$ **do**
    for each *flow* $v_i \in \Gamma$ **do**
      $J_i^h = Smax_i^h - Smin_i^h$
    end
    $\mathcal{B}^h = \mathcal{BP}^h + \max_{v_j \in \Gamma_h}\left\{\left(1+\left\lfloor\frac{J_j^h}{T_j}\right\rfloor\right)\cdot T_j - J_j^h\right\}$
    $W^h(t) = \sum_{v_j \in \Gamma_h} RBF_j^h(t)$
    for each *flow* $v_i \in \Gamma_h$ **do**
      $Bklg_i^h = \max_{0 \le t \le \mathcal{B}^h}\{W^h(t) - C_i - t\}$
      if $h \neq last_i$ **then**
        $Smin_i^{h+1} = Smin_i^h + C_i + L$
        $Smax_i^{h+1} = Smax_i^h + Bklg_i^h + C_i + L$
      else
        $R_i = Smax_i^h + Bklg_i^h + C_i$
      end
    end
  end
end
Output: R_i for each flow $v_i \in \Gamma$.

with $W^h(t)$ being defined by Eq. (4). The computation ends with $\mathcal{B}^h$, which is obtained using Eq. (19).

The pseudo-code for the FA approach is given using Algorithm 1. In fact, computation can be done in a discrete manner by testing time t only at arrival times of the new packets.[10] This computation does not consider the serialization effect.

5. Case Study and Comparison

A comparative analysis is performed for both sample AFDX configurations and real avionics transmission systems. We test different approaches including the Simulation approach, the Model Checking approach, the Network Calculus approach, the Trajectory approach and the FA approach.

5.1. Delays on sample AFDX configurations

An illustrative avionics transmission system is depicted in Fig. 6 and the corresponding configuration is abstracted as the network in Fig. 7. The flow characteristics are given in Table 1. As an example, we analyze flow v_1 ($\mathcal{P}_1 = e_1$–S_1–S_3–e_6), which meets flow v_2 at switch S_1, and meets v_3, v_4 and v_5 at switch S_3.

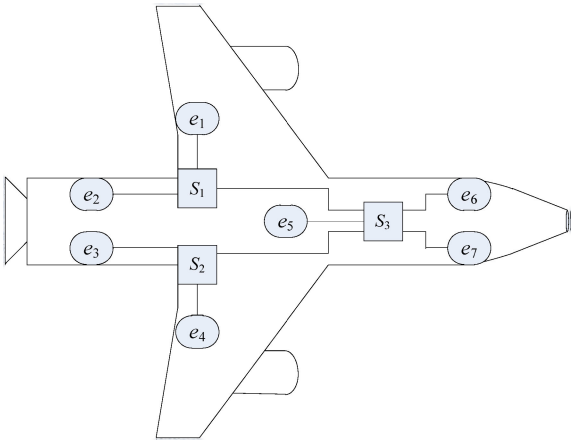


Fig. 6. A sample transmission system.

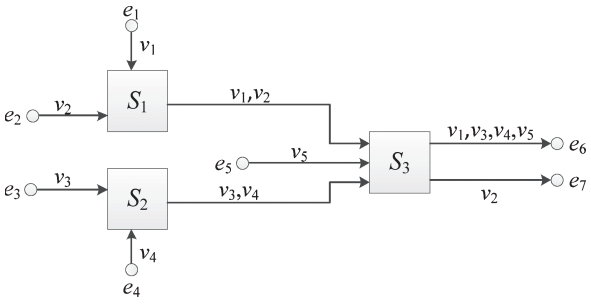


Fig. 7. Network topology of the sample system.

Table 1. Characteristics of flows.

Sender	Receiver	Path	Packet length (bits)	Period (µs)
e_1	e_6	e_1–S_1–S_3–e_6	4,000	4,000
e_2	e_7	e_2–S_1–S_3–e_7	4,000	4,000
e_3	e_6	e_3–S_2–S_3–e_6	4,000	4,000
e_4	e_6	e_4–S_2–S_3–e_6	4,000	4,000
e_5	e_6	e_5–S_3–e_6	4,000	4,000

We need to analyze all of the nodes along $\mathcal{P}_1$, as described in Algorithm 1. By definition, we have $Smin_1^{e_1} = Smax_1^{e_1} = 0$. As v_1 is the only flow emitted from e_1, the maximum backlog can be computed using:

$$Bklg_1^{e_1} = \max_{0 \le t \le \mathcal{B}^{e_1}}\left\{\left(1+\left\lfloor\frac{t}{T_1}\right\rfloor\right)\cdot C_1 - C_1 - t\right\}$$

According to Lemmas 4 and 5, we have:

$$\mathcal{BP}^{e_1} = C_1 = 40\,\mu s$$

$$\mathcal{BS}^{e_1} = \left(1+\left\lfloor\frac{0}{T_1}\right\rfloor\right)\cdot T_1 = 4{,}000\,\mu s$$

where the arrival jitter is $J^{e_1} = 0$.

The test interval is obtained as:

$$\mathcal{B}^{e_1} = \mathcal{BP}^{e_1} + \mathcal{BS}^{e_1} = 40 + 4{,}000 = 4{,}040\,\mu s$$

©2021 JSASS

Table 2. Durations to reach node S_3.

	v_1	v_3	v_4	v_5
$Smin_j^{S_3}$	112	112	112	56
$Smax_j^{S_3}$	152	152	152	56
$J_j^{S_3}$	40	40	40	0

The maximum backlog is obtained at time $t = 0$, so that $Bklg_1^{e_1} = 0$. According to Eq. (1), $Smax_1^{S_1} = 56\,\mu s$. With Eq. (3), we obtain $Smin_1^{S_1} = 40 + 16 = 56\,\mu s$. The arrival jitter is 0. Similarly, $Smax_2^{S_1} = Smax_2^{S_1} = 56\,\mu s$, and the arrival jitter is 0.

At node S_1, flow v_1 meets v_2, and the maximum backlog $Bklg_1^{S_1}$ is expressed as:

$$Bklg_1^{S_1} = \max_{0 \le t \le \mathcal{B}^{S_1}} \left\{ \left(1 + \left\lfloor \frac{t}{T_1} \right\rfloor\right) \cdot C_1 + \left(1 + \left\lfloor \frac{t}{T_2} \right\rfloor\right) \cdot C_2 - C_1 - t \right\}$$

According to Lemmas 4 and 5, we have:

$$\mathcal{BP}^{S_1} = C_1 + C_2 = 80\,\mu s$$

$$\mathcal{BS}^{S_1} = \left(1 + \left\lfloor \frac{0}{T_1} \right\rfloor\right) \cdot T_1 = 4{,}000\,\mu s$$

where the arrival jitter is $J^{S_1} = 0$.

The test interval is obtained as:

$$\mathcal{B}^{S_1} = \mathcal{BP}^{S_1} + \mathcal{BS}^{S_1} = 80 + 4{,}000 = 4{,}080\,\mu s$$

The maximum backlog is obtained at time $t = 0$, where $Bklg_1^{S_1} = 40\,\mu s$ and $Smax_1^{S_3} = 56 + 40 + 40 + 16 = 152\,\mu s$.

Finally, in the last node of $\mathcal{P}_1$, flow v_1 meets v_3, v_4 and v_5. The values of $Smax_j^{S_3}$ and $Smin_j^{S_3}$ for these four flows are depicted in Table 2.

The maximum backlog $Bklg_1^{S_3}$ can be expressed as:

$$Bklg_1^{S_3} = \max_{0 \le t \le \mathcal{B}^{S_3}} \left\{ \left(1 + \left\lfloor \frac{t + J_1^{S_3}}{T_1} \right\rfloor\right) \cdot C_1 + \left(1 + \left\lfloor \frac{t + J_3^{S_3}}{T_3} \right\rfloor\right) \cdot C_3 + \left(1 + \left\lfloor \frac{t + J_4^{S_3}}{T_4} \right\rfloor\right) \cdot C_4 + \left(1 + \left\lfloor \frac{t + J_5^{S_3}}{T_5} \right\rfloor\right) \cdot C_5 - C_1 - t \right\}$$

According to Lemmas 4 and 5, we have:

$$\mathcal{BP}^{S_3} = C_1 + C_3 + C_4 + C_5 = 160\,\mu s$$

$$\mathcal{BS}^{S_3} = \left(1 + \left\lfloor \frac{0 + 40}{T_1} \right\rfloor\right) \cdot T_1 = 4{,}000\,\mu s$$

The test interval is obtained as:

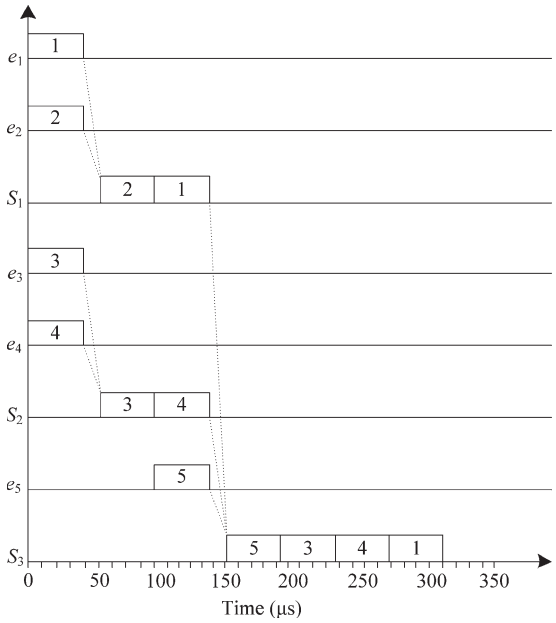


Fig. 8. Worst-case scenario corresponding to the FA approach.

Table 3. Worst-case end-to-end transmission delays.

	Simulation	Model Checking	Network Calculus	Trajectory Approach	FA
v_1	203.7	272	313.2	312	312
v_2	178.4	192	192.4	192	192
v_3	217.1	272	313.2	272	312
v_4	183.8	272	313.2	272	312
v_5	125.7	176	217.2	216	216

$$\mathcal{B}^{S_3} = \mathcal{BP}^{S_3} + \mathcal{BS}^{S_3} = 160 + 4{,}000 = 4{,}160\,\mu s$$

The maximum backlog encountered by flow v_1 at node S_3 is obtained at time $t = 0$ and $Bklg_1^{S_3} = 120\,\mu s$. The worst-case end-to-end transmission delay of flow v_1 is $R_1 = Smax_1^{S_3} + Bklg_1^{S_3} + C_1 = 152 + 120 + 40 = 312\,\mu s$. The corresponding worst-case transmission scenario is depicted in Fig. 8.

Similarly, the worst-case end-to-end transmission delays of all of the flows in Table 1 are given in Table 3.

From Table 3, we see that the Simulation approach only obtains an average delay due to missing some rare cases. For all five flows, both the FA and Trajectory approaches compute a tighter delay upper boundary than the Network Calculus approach. For flows v_1 and v_5, the FA and Trajectory approaches compute the same delay upper boundary, which has 40 μs pessimism. For flow v_2, the two approaches can compute the exact delay upper boundary. For flows v_3 and v_4, the FA approach computes a pessimistic upper boundary (i.e., 40 μs overestimated) compared to the exact boundary computed using the Model Checking approach, but the Trajectory approach can compute the exact delay upper boundary without any pessimism.

From the network depicted in Fig. 7, packets of flows v_3 and v_4 are transmitted through a common physical link,

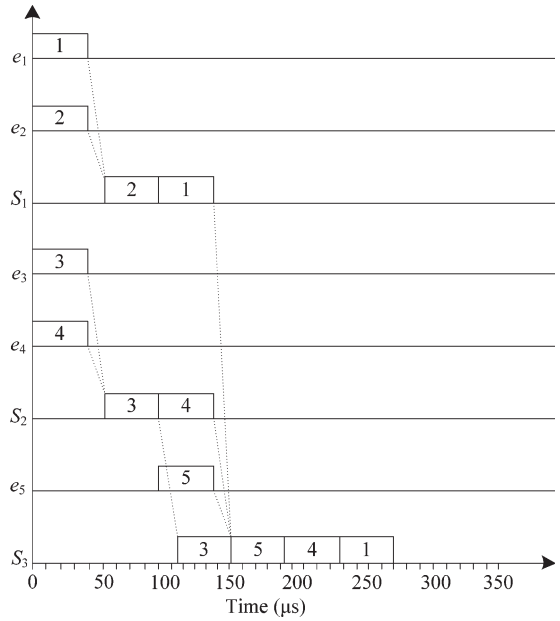


Fig. 9. Actual worst-case scenario for packets of flow v_1.

Table 4. BAGs and packet lengths.

BAG (μs)	Number of VLs	Packet length (bits)	Number of VLs
2,000	193	1,501–2,000	187
2,500	144	2,001–2,500	139
3,000	187	2,501–3,000	167
3,500	158	3,001–3,500	235
4,000	211	3,501–4,000	165

S_2–S_3, meeting flow v_1 at node S_3. In fact, packets of flows v_3 and v_4 cannot simultaneously arrive at node S_3. They can only arrive one after another, as depicted in Fig. 9. However, the FA approach considers that they arrive simultaneously when computing $Bklg_1^{S_3}$, and that is why 40-μs pessimism is induced. The Trajectory approach assumes that the packets of each competing flow delay the packet under study only once throughout the entire transmission period. For flows v_3 and v_4, this assumption can avoid the pessimism caused by the FA approach. For some special cases such as flows v_3 and v_4 in Fig. 7, the FA approach can induce more pessimism than the Trajectory approach when computing an end-to-end delay upper boundary. A short description of the Trajectory approach and the corresponding computation is added at the end of this paper as Appendix.

5.2. Delays on real avionics systems

This section validates using the FA approach for a real avionics network that includes 98 end systems, 16 switches and 893 virtual links.

In Table 4, the left side demonstrates the dispatching of VLs among different BAGs and the right side demonstrates the dispatching of VLs among different packet lengths. Table 5 gives the number of VL paths according to length (i.e., number of crossed switches). The maximum path length is five.

As the Model Checking approach has the combinational explosion problem when used in real industrial avionics networks with large configurations, we only compare the Network Calculus approach and FA approach. For each flow v_i, the result obtained using the Network Calculus approach, NC_i, is taken as the reference value and normalized to 100. The result obtained using the FA approach, FA_i, is normalized in a similar way as below:

$$N_FA_i = 100 + \left(\frac{FA_i - NC_i}{NC_i}\right) \times 100$$

This experiment is performed using different packet lengths. The VLs are sorted by the increasing order of N_FA_i. All of the values are depicted in Fig. 10.

Neither the FA approach nor the Network Calculus approach induce optimism, and from the figure above, we can see that the FA approach can compute a tighter upper boundary than the Network Calculus approach when conducting more than 2,000 tests. Since we don't know the exact worst-case end-to-end delay, currently it is not possible to evaluate the pessimism caused when using the FA approach.

Table 5. Length of path.

Number of crossed switches	Number of paths
1	145
2	237
3	256
4	157
5	98

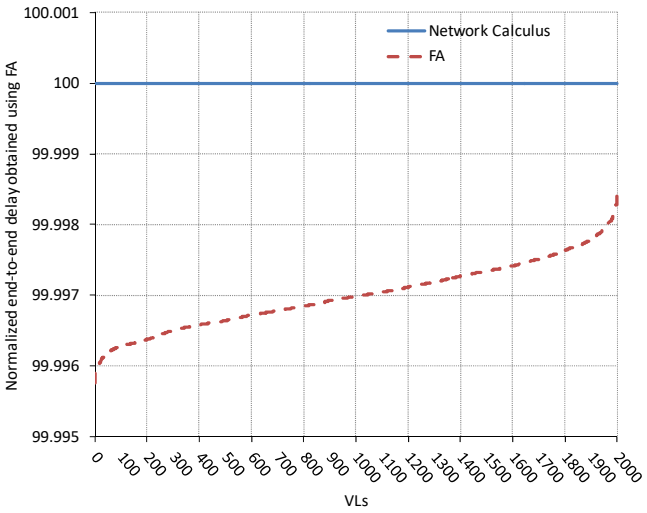


Fig. 10. Comparison results of the two approaches.

6. Conclusion

Transmission performance is extremely important for the avionics networks of aircraft. It is very relevant to avionics systems such as safety-critical systems like flight control. A deterministic upper boundary of the end-to-end transmission delay of avionics flows needs to be guaranteed, and the upper boundary is usually used to evaluate the transmission performance of avionics networks.

This paper introduces the Forward end-to-end delay Analysis approach, which evaluates transmission performance by computing the delay upper boundaries of flows in avionics real-time networks such as the Avionics Full DupleX switched Ethernet (AFDX).

We tested this approach on both sample AFDX networks and real industrial avionics systems. The FA approach does not have the problem of combinational explosion for networks with large configurations. We compared the results obtained using the FA and Network Calculus approaches, and the comparative analysis conducted shows that the FA approach obtains a tighter upper boundary than the Network Calculus approach. We also compared the FA and Trajectory approaches, and presented the reasons why the FA approach has more pessimism than the Trajectory approach for some special cases.

In future research, we will try to reduce the pessimism in the FA approach. Additionally, the FIFO scheduling may not be suitable, so other service disciplines such as static priority queuing should also be considered.

Acknowledgments

This paper was written with the help of the CPA&DCS Bus Test and Validation Laboratory of Shanghai Aviation Electric Co., Ltd. The authors would like to thank Zhangjin Li, Jizhong Liu and his research team for their inspirational discussions.

The data used to support the findings of this study are included within the article.

The authors declare that there are no conflicts of interest regarding the publication of this paper.

This work is sponsored by the China 1000 Young Talents Program, the Young Talent Support Plan of Xi'an Jiaotong University, NSF China (NSFC) under Grant Nos. 61772410 and 61572398. The work is also sponsored by the China Post-Doctoral Science Foundation under Grant No. 2015M570837 and the National Natural Science Foundation of China (NSFC) under Grant No. 61502381.

References

1) Huang, W. N., Zhang, J. M., and Liu, X.: Transmission Performance of the Mechanical Control System in the Aircraft, *Appl. Mech. Mater.*, **484-485** (2014), pp. 368–372. DOI: 10.4028/www.scientific.net/AMM.484-485.368
2) Wan, S., Guo, J., and Wang, W.: Research and Testing on Transmission Performance of Aircraft Radome, *J. Nanjing Univ. Aeronautics Astronautics*, **41** (2009), pp. 57–61.
3) Moir, I. and Seabridge, A.: Aircraft System Examples, *Design and Development of Aircraft Systems*, **12** (2012), pp. 265–304. DOI: 10.1002/9781118469156.ch10
4) Song, S., Weiguo, L., and Peitsch, D.: Detailed Design of a High Speed Switched Reluctance Starter/Generator for More/All Electric Aircraft, *Chinese J. Aeronautics*, **23**, 2 (2010), pp. 216–226. DOI: 10.1016/S1000-9361(09)60208-9
5) ARINC 664, Aircraft Data Network, Part 7, Technical Report, ARINC Specification 664, 2005.
6) Wang, G., Bo, Y., and Yang, Q.: Key Technologies of ARINC664 Bus Testing, Fifth International Conference on Instrumentation & Measurement, Sep., 2016. DOI: 10.1109/IMCCC.2015.27
7) Liu, Y., Wang, H., and Wang, B.: Design and Implementation of AFDX End System Software, *Electronics Optics & Control*, **19**, 11 (2012), pp. 71–76.
8) Chen, D. and Song, D.: AFDX Network Performance Testing, Proceedings of the First Symposium on Aviation Maintenance and Management, 2014, pp. 313–322. DOI: 10.1007/978-3-642-54236-7_35
9) Suthaputchakun, C., Lee, K. M. B., and Sun, Z.: Impact of End System Scheduling Policies on AFDX Performance in Avionic On-board Data Network, International Conference on Advanced Informatics: Concepts, Aug., 2015. DOI: 10.1109/ICAICTA.2015.7335388
10) Kemayo, G., Ridouard, F., and Bauer, H.: A Forward End-to-end Delays Analysis for Packet Switched Networks, International Conference on Real-time Networks & Systems, 2014, pp. 55–74. DOI: 10.1145/2659787.2659801
11) Kemayo, G., Benammar, N., and Ridouard, F.: Improving AFDX End-to-end Delays Analysis, IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Sep., 2015. DOI: 10.1109/ETFA.2015.7301463
12) Coelho, R., Fohler, G., and Scharbarg, J. L.: Upper Bound Computation for Buffer Backlog on AFDX Networks with Multiple Priority Virtual Links, Symposium on Applied Computing, 2017. DOI: 10.1145/3019612.3019729
13) Baga, Y., Ghaffari, F., and Zante, E.: Worst Frame Backlog Estimation in an Avionics Full-duplex Switched Ethernet End-system, Digital Avionics Systems Conference, Sep., 2016. DOI: 10.1109/DASC.2016.7777990
14) Baga, Y., Richaud, M., and Ghaffari, F.: Probabilistic Model of AFDX Frames Reception for End System Backlog Assessment, IEEE International Symposium on Industrial Embedded Systems, Feb., 2017. DOI: 10.1109/SIES.2017.7993400
15) Ashjaei, M., Behnam, M., and Nolte, T.: SEtSim: A Modular Simulation Tool for Switched Ethernet Networks, *J. Syst. Architecture*, **65** (2016), pp. 1–14. DOI: 10.1016/j.sysarc.2016.02.002
16) Fernandez-Olmos, L., Burrull, F., and Pavon-Marino, P.: Net2Plan-AFDX: An Open-source Tool for Optimization and Performance Evaluation of AFDX Networks, IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 2016, pp. 1–7. DOI: 10.1109/DASC.2016.7778026
17) Dong, S., Zeng, X., and Ding, L.: The Design and Implementation of the AFDX Network Simulation System, International Conference on Multimedia Technology, 2010, pp. 1–4.
18) Adnan, M., Scharbarg, J. L., and Ermont, J. R. M.: Model for Worst Case Delay Analysis of an AFDX Network Using Timed Automata, IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010), Oct., 2010. DOI: 10.1109/ETFA.2010.5641124
19) Adnan, M., Scharbarg, J. L., and Ermont, J.: An Improved Timed Automata Model for Computing Exact Worst-case Delays of AFDX Periodic Flows, IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012, pp. 1–4.
20) Cruz, R. L.: A Calculus for Network Delay. I. Network Elements in Isolation, *IEEE Trans. Information Theory*, **37**, 1 (1991), pp. 114–131. DOI: 10.1109/18.61109
21) Cruz, R. L.: A Calculus for Network Delay, Part II: Network Analysis, *IEEE Trans. Information Theory*, **37** (1991), pp. 132–141. DOI: 10.1109/18.61110
22) Jing, X. and Min, X.: Delay Bound Analysis in Real-time Networks with Priority Scheduling Using Network Calculus, IEEE International Conference on Communications, 2013, pp. 2469–2474. DOI: 10.1109/ICC.2013.6654903
23) Xu, Q. and Yang, X.: Performance Analysis on Transmission Estimation for Avionics Real-Time System Using Optimized Network Calculus, *Int. J. Aeronaut. Space Sci.*, (2019). DOI: 10.1007/s42405-018-00140-7
24) Feng, H. E. and Ershuai, L. I.: Deterministic Bound for Avionics Switched Networks According to Networking Features Using Network Calculus, *Chinese J. Aeronautics*, **30**, 6 (2017). DOI: 10.1016/j.cja.2017.08.010
25) Bauer, H., Scharbarg, J. L., and Fraboul, C.: Worst-case End-to-end Delay Analysis of an Avionics AFDX Network, Design, Automation & Test in Europe Conference & Exhibition, 2010, pp. 1220–1224. DOI: 10.1109/DATE.2010.5456993
26) Bauer, H., Scharbarg, J. L., and Fraboul, C.: Applying and Optimizing Trajectory Approach for Performance Evaluation of AFDX Avionics Network, IEEE International Conference on Emerging Technologies & Factory Automation, Oct., 2009. DOI: 10.1109/ETFA.2009.5347083

©2021 JSASS

27) Li, X., Cros, O., and George, L.: The Trajectory Approach for AFDX FIFO Networks Revisited and Corrected, IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications, Aug., 2014. DOI: 10.1109/RTCSA.2014.6910523
28) Martin, S. and Minet, P.: Schedulability Analysis of Flows Scheduled with FIFO: Application to the Expedited Forwarding Class, International Parallel & Distributed Processing Symposium, 2006, pp. 167–180.
29) Bauer, H., Scharbarg, J. L., and Fraboul, C.: Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach, *IEEE Trans. Industrial Informatics*, **6**, 4 (2010), pp. 521–533. DOI: 10.1109/tii.2010.2055877
30) Kemayo, G., Ridouard, F., and Bauer, H.: Optimism Due to Serialization in the Trajectory Approach for Switched Ethernet Networks, 7th Junior Researcher Workshop on Real-Time Computing, Sep., 2013, pp. 1–8.
31) Kemayo, G., Ridouard, F., and Bauer, H.: Optimistic Problems in the Trajectory Approach in FIFO Context, IEEE Conference on Emerging Technologies & Factory Automation, 2013, pp. 1–8. DOI: 10.1109/ETFA.2013.6648054
32) Zhen, D., Feng, H. E., and Zhang, Y.: Real-time Path Optimization Algorithm of AFDX Virtual Link, *Acta Aeronautica Astronautica Sinica*, (2015). DOI: 10.7527/S1000-6893.2014.0323
33) Cha, Y. J., Lim, J. H., and Lee, S. Y.: New Feasible Grouping Algorithms for Virtual Link in AFDX Networks, International Conference on Informatics, Jun., 2015. DOI: 10.1109/ICIEV.2015.7334035
34) Scharbarg, J. L., Ridouard, F., and Fraboul, C.: A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network, *IEEE Trans. Industrial Informatics*, **5**, 1 (2009), pp. 38–49. DOI: 10.1109/TII.2009.2016085
35) Benammar, N., Ridouard, F., and Bauer, H.: Forward End-To-End Delay Analysis for AFDX Networks, *IEEE Trans. Industrial Informatics*, **14**, 3 (2017), pp. 858–865. DOI: 10.1109/TII.2017.2720799

Appendix

Trajectory approach

The Trajectory approach is designed to compute an upper boundary of the end-to-end delays for flows transmitted over an AFDX network. For flow v_i, it computes the latest starting time $W_i^{last_i}(t)$ at the last visited switch. The latest starting time $W_i^{last_i}(t)$ is computed as shown below:

$$W_i^{last_i}(t) = \sum_{\substack{j\in[1,n]\\ j\neq i}} \left(1 + \left\lfloor \frac{t + A_{i,j}}{T_j} \right\rfloor\right) \cdot C_j \tag{a}$$

$$+ \left(1 + \left\lfloor \frac{t}{T_i} \right\rfloor\right) \cdot C_i \tag{b}$$

$$+ \sum_{\substack{h_k\in\mathcal{P}_i\\ h_k\neq last_i}} \left(\max_{\substack{j\in[1,n],\\ h_k\in\mathcal{P}_j}} (C_j)\right) \tag{c}$$

$$+ (|\mathcal{P}_i| - 1) \cdot L \tag{d}$$

$$- \sum_{\substack{h_k\in\mathcal{P}_i\\ h_k\neq first_i}} (\Delta_i^k) \tag{e}$$

$$- C_i \tag{f}$$

Term (a) corresponds to the processing time of packets from every flow v_j crossing flow v_i and transmitted in the same busy period as the considered packet. $A_{i,j}$ integrates the maximum jitter of packets from v_i and v_j on their first shared output port.

Term (b) is the packet processing time (i.e., on one node) of flow v_i being transmitted in the same busy period as the considered packet.

Term (c) is the processing time of the longest packet for each node in path $\mathcal{P}_i$, except for the last one. It represents the packets that must be counted twice.

Term (d) corresponds to the sum of switching delay.

Term (e) sums the duration between the beginning of the busy period and the arrival of the first packet coming from the preceding node in path $\mathcal{P}_i$ for each node h.

Term (f) presents that C_i is subtracted, as $W_{i,t}^{last_i}$ is the latest starting time instead of the ending time for the considered packet at the last node along path $\mathcal{P}_i$.

For flow v_i, an upper boundary R_i of the end-to-end transmission delays is obtained with

$$R_i = \max(W_i^{last_i}(t) + C_i - t).$$

Cees Bil
Associate Editor