

# Using Holistic Queue Scheduling to Optimize the Real-Time Performance of AFDX

Liu Jing<sup>1,2,\*</sup>

<sup>1</sup>Institute of Mathematics and Information Science, Weinan Normal University, Weinan 714000, China

<sup>2</sup>Research center of Weinan Wisdom City Engineering Technology, Weinan Normal University, Weinan 714000, China

**Abstract:** AFDX, Avionics Full Duplex Switched Ethernet Network, is an aircraft data network using the FIFO scheduling strategy to transmit data. In this circumstance, the delay of the critical data is large, while congestion is uncertain. To cope with this problem, this paper presents a Holistic Queue Scheduling policy to optimize the real-time performance of AFDX, and estimate the latency bounds based on Network Calculus. Simulation results show that this policy not only ensures real-time critical data, but also ensures the fairness of scheduling non-critical data. And this policy can reduce the impact of malicious data in some extent.

**Keywords:** Holistic Queue Scheduling, Network Calculus, Avionics Full Duplex Switched Ethernet Network, real-time.

## 1. INTRODUCTION

In recent years, the avionics system network protocols, an important part of modern avionics system, has got the development by leaps and bounds [1]. Many ADNs (Aircraft Data Networks) are proposed with a high available bandwidth, minimum wiring to reduce weight, and low the development costs, such as ARINC (Aeronautical Radio Incorporated) 429, MIL-STD-1553 (Military-Standard-1553) and ARINC 629 with a bandwidth of 100Kbps, 1Mbps and 2Mbps. For the new generation Airbus A380, none of the existing ADNs could fulfill the requirement. As a consequence, the Avionics Full Duplex switched Ethernet (AFDX) was conceived by Airbus based on the IEEE 802.3 [2] and ARINC 664 [3] and implemented on the A380 for the first time. The AFDX network adopts full-duplex Ethernet technology to avoid the resource conflicts in the End-System. However, since a variety of data streams compete for resources in the switch resulting in a greater latency, jitter and unfairness, the AFDX network does not completely eliminate uncertainty of traditional Ethernet transmission [4].

Many researchers study on the performance and reliability verification of the AFDX network. The End-to-End delay is the key parameter in practical application. The AFDX protocol uses First-In First-Out (FIFO) algorithm to transmit messages. The FIFO algorithm forwards the data according the order they arrive and regardless the priorities of the data flows. H. Charara et al.<sup>[4]</sup> and J. Scharbarg *et al.* [7] calculated the delay bound of the AFDX network by network calculus [5-6], discovering that the latency is less than the delay bound and FIFO algorithm cannot fulfill the requirements of urgent messages.

In order to fulfill the requirements of different flows, many algorithms are used in the AFDX network. F. Ri-douard *et al.* [8] and X. Chen *et al.* [9] used Static Priority (SP) to reduce the delay bound of the safety-critical data flows and urgent messages, but without considering the fairness of flows with different priorities. H. Dong *et al.* [10] reduced the delay bound of some smaller non-safety-critical messages using load classification, but it leads to a greater latency of bigger messages. Y. Hua *et al.* [11] used Deficit Round Robin (DRR) algorithm to ensure the fairness of data flow, but it enlarges the delay bound of the safety-critical data flows.

Based on analysis of different flows' real-time requirements in the AFDX network, we proposed a new scheduling policy that ensures the real-time requirement and provides fair scheduling service to the non-safety-critical data flows. And under the specific AFDX network configuration, it shows the function of new scheduling policy in AFDX network real-time optimization from two aspects of theoretical derivation and experimental simulation. We deduced the delay bound drawing on the network calculus and simulated the AFDX network with the new policy.

## 2. CONTEXT OF STUDY

Here is a brief overview of AFDX network, network calculus and the multiplexer.

### A. the AFDX network

Compared with the characteristics of Ethernet, previous types of ADNs cannot adapt to the rapid development in the avionics requirements. In order to cope with this problem, the AFDX network has been defined in the part 7 of ARINC 664 specification and has become the reference technology of avionics. The AFDX is a 100Mbps switched Ethernet network on which flows are pre-defined.

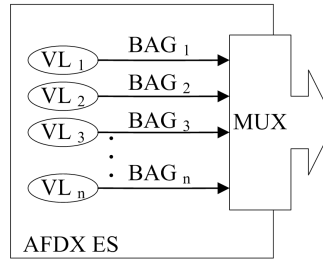


Fig. (1). End-System output model.

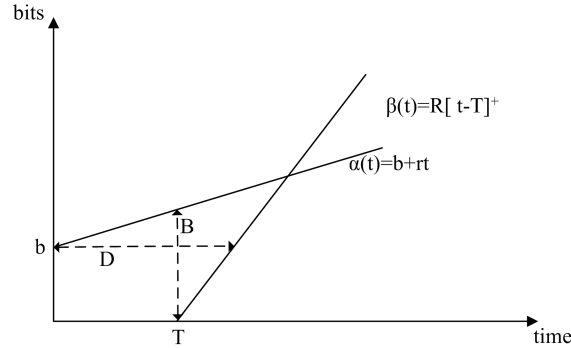


Fig. (2). Delay and backlog through the network element.

Since conventional Ethernet is a non-deterministic network, AFDX has to be extended to ensure real-time quality and a high reliability in order to comply with the strict requirements of ADNs. AFDX ensures real-time quality through traffic control. Traffic control is achieved by guaranteeing the bandwidth of each logical channel, called a Virtual Link (VL), thus limiting the jitter and transmit latency. AFDX consists of End-Systems (ESs), switches and physical links. Traffic control in VLs is guaranteed by ES.

As shown in Fig. (1), each AFDX ES output capability is limited by the multiplexer (MUX) in order to guarantee their Bandwidth Allocation Gap (BAG).

### B. Network Calculus

In the early nineties, R. L. Cruz analyzed the quality of service of six network elements, and defined the conception of the shaper, service curve and arrival curve. Subsequently, J. Y. Boudec and C.S. Chang developed the theory of network calculus by introducing a mathematical representation of min-plus algebra. Then it becomes popular, while the stochastic network calculus has also made some studies [13]. For a network element, the quantum of arrived data is less than arrival curve, and the element can forward the data more than service curve. We can calculate the bound of delay, backlog and other parameters by the network calculus theory with arrival curve and service curve easily.

As shown in Fig. (2), the delay of a bit is bounded by the horizontal distance between arrival curve  $\alpha$  and service curve  $\beta$ , while the maximum backlog  $B$  of the element is bounded by the vertical distance.

### C. the Multiplexer

The multiplexer (MUX) has two or more input links and a single output link. It is used for simulating scheduling of a switch. In this paper, we analyze multiplexers with three non-preemptive service algorithms (FIFO, SP and DRR). In our model, the MUX has some buffers in a store-and-forward way, and it is a work-conserving system. Non-preemptive means that once a frame begins transmit on the output link, the transmission is not allowed to be interrupted by any other flows regardless of their priorities.

Now we assume  $R$  as the rate of the stream, while letting  $R$  take on a value between 0 and the transmission capacity of the link. For example, if the transmission capacity of the link is  $C$ , then  $R(t) \in [0, C]$ . The rate of input link  $i$  is  $R_i$ , and the maximum transmission rate of input link  $i$  is  $C_i$ . The rate of output link is  $R_{out}$ , and its transmission capacity is  $C_{out}$ .

#### 1) FIFO MUX

The first-in first-out multiplexer (FIFO MUX) contains one buffer and transmits the frame in the order they arrived (Fig. 3). We assume  $B(t)$  as the backlog of the buffer which is in the front of the FIFO MUX at time  $t$ .

If we assume the size of the longest frame is  $L_{max}$ , the maximum delay of a frame with the length  $L$  bytes is

$$D(t) = \frac{L + B(t)}{C_{out}}, \text{ while } B(t) \geq L_{max}.$$

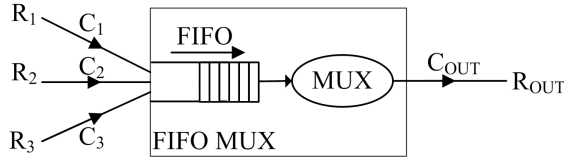


Fig. (3). FIFO MUX with one buffer.

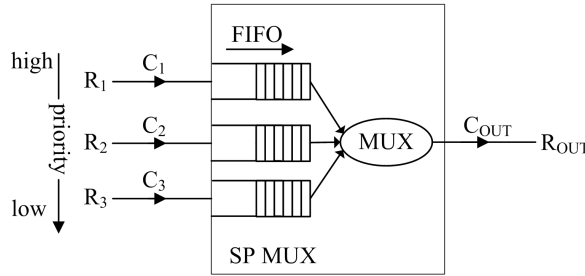


Fig. (4). SP MUX with multi-buffers.

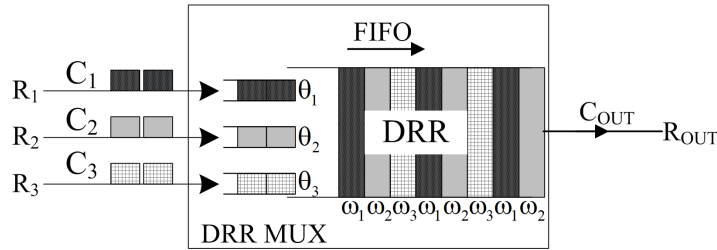


Fig. (5). DRR MUX with multi-buffers.

2) SP MUX

We analyze a Static Priority multiplexer (SP MUX) with three priority classes, while flows in the input link 1 have the highest priority, flows in the input link 2 have the second priority, and flows in the input link 3 have the lowest priority (Fig. 4). We assume  $B_i(t)$  as the backlog of the buffer  $i$  in the front of the SP MUX at time  $t$ .

We can deduce the maximum delay of a frame from stream  $i$  with the length  $L$  bytes arrive at time  $t$  is

$$D = \frac{\sum_{j=1}^{i-1} B_j(t) + L}{C_{out} - \sum_{j=1}^{i-1} C_j}$$

3) DRR MUX

In the same way, we analyze a Deficit Round Robin multiplexer (DRR MUX) with three flows, while flows in the input link  $i$  is endowed with a counter  $\theta_i$  and a allocated bandwidth  $\omega_i$  (Fig. 5). It dispatches data in bytes. In a polling cycle, it updates counter of each link  $\theta_i = \theta_i + \omega_i$ . If the length of the frame  $l_i$  in the input link  $i$  is less than the counter, the frame is transmitted, while update counter

$\theta_i = \theta_i - l_i$ . After updating the counter, then it starts to schedule the next input link.

We can deduce the maximum delay of a frame from stream  $i$  with the length  $L$  bytes arrived at time  $t$  is

$$D_i = \frac{\sum_{j=1}^n \omega_j}{C_{out}} \left( \frac{L}{\omega_i} + \left\lceil \frac{L_{max}}{\omega_i} \right\rceil \right)$$

3. HOLISTIC QUEUE SCHEDULING ON AFDX

There are four important types of flows identified by their periodicities and temporal deadlines in the avionics system [12]: (a) Urgent sporadic messages that have to be transmitted under a predefined small bounded time, such as alarms; (b) Periodic messages that are also strongly time constrained, such as sensor data; (c) Sporadic messages that have known deadlines to respect but without any urgency; (d) Sporadic messages that do not have to respect strict time constraints, such as media file. We analyze the first three types of flows with real-time requirement. We can use SP algorithm to reduce the transmission delay of flows with higher priority effectively. However, the fairness of different flows is important.

Based on the analysis of real-time requirement in the AFDX network, we proposed to combine three different al-

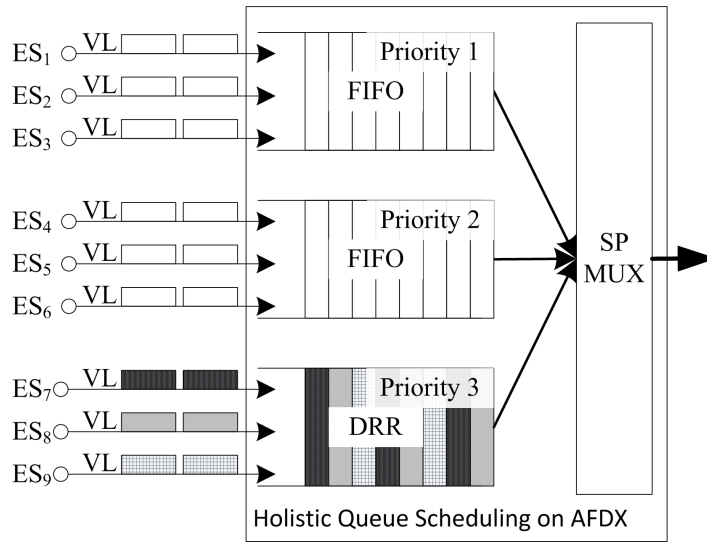


Fig. (6). AFDX logical structure with HQSA.

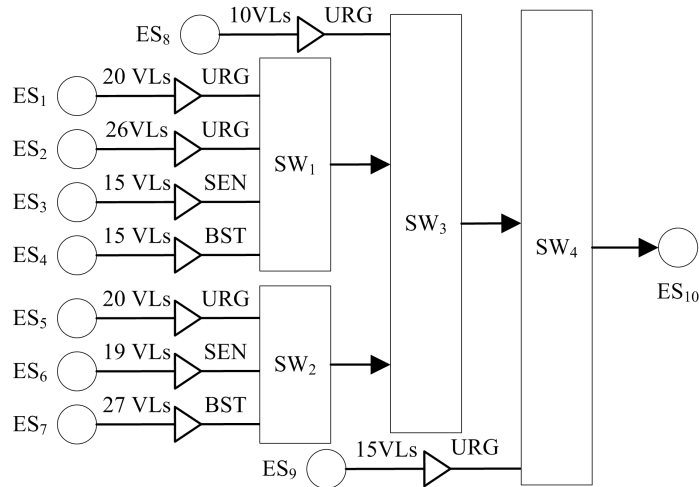


Fig. (7). AFDX prototype configuration.

gorithms (FIFO, SP and DRR) to schedule the flows. This policy is used for flows in the AFDX network, so it named as Holistic Queue Scheduling on AFDX (HQSA).

As shown in Fig. (6), HQSA gives the flows three different priorities, and serves them in SP algorithm. Urgent sporadic messages and periodic messages are safety-critical data flows, and are endowed with higher priority. Sporadic messages that have known deadlines to respect but without any urgency are non-safety-critical data flow, and are endowed with the lowest priority. For the same priority, safety-critical data flows are served in FIFO algorithm, while non-safety-critical data flows are served in DRR algorithm. In addition, the switch is a work-conserving system that keeps in the state of non-preemptive.

In an AFDX End-System (ES), virtual link (VL) mechanism is used to ensure real-time demand, and multiple VLs can coexist in the same physical link. There are two essential

parameters in VL: (1) the minimum Bandwidth Allocation Gap (BAG) between two adjacent frames, which ranges from 1-128ms; (2) the maximum length of the VL allows the transmission frame  $L_{max}$ .

We modeled an AFDX network with HQSA to analyze the performance of the policy. For simplicity of modeling, we didn't analyze the redundancy in the model, and the buffer of the switch is large enough to store the frames.

As shown in Fig. (7), the case is a tree topology with multi-levels. There are 4 switches ( $SW_1, SW_2, SW_3, SW_4$ ), 9 source End-Systems ( $ES_1, ES_2, \dots, ES_9$ ) and 1 destination End-System ( $ES_{10}$ ). Every physical link has multi VLs and the length of the link is 10 meters. The switches schedule data flows in HQSA policy while transmission capacity is  $C = 100 Mb/s$ . All links in this model are static, so that there is no latency for searching route path.

**Table 1. Parameter settings for BAG and frame length.**

URG VLs			
BAG(ms)	Number of VLs	Frame Length (bytes)	Number of VLs
2	2	0-100	40
4	3	100-200	21
8	6	200-400	13
16	11	400-600	8
32	26	600-800	5
64	23	800-1000	3
128	20	>1000	1

SEN VLs			
BAG(ms)	Number of VLs	Frame Length (bytes)	Number of VLs
2	1	0-100	3
4	2	100-200	6
8	2	200-400	10
16	5	400-600	7
32	11	600-800	5
64	7	800-1000	3
128	6	>1000	1

BST VLs			
BAG(ms)	Number of VLs	Frame Length (bytes)	Number of VLs
2	2	0-100	3
4	3	100-200	4
8	4	200-400	7
16	7	400-600	16
32	17	600-800	9
64	11	800-1000	7
128	8	>1000	6

We assume data flows with 1st priority as urgent data flows (URG), data flows with 2nd priority as sensor data flows (SEN), and data flows with 3rd priority as best effort data flows (BST). The parameter settings in various types of VL are shown in Table 1.

To verify the fairness of HQSA policy, we assume 2 unreasonable data flows, whose length is set as 1518 bytes and

BAG is 2ms from ES<sub>4</sub> and ES<sub>7</sub>. Since  $1518 \times 8 / 0.002 = 12144000$  b/s, 12144000 b/s is occupied by these two data flows. The situation should be avoided, so we call them malicious data flows.

There are several physical cables, whose transmission rate is a fixed value. So we analyze it as the constant delay element as defined by R. L. Cruz [5], and its service curve is

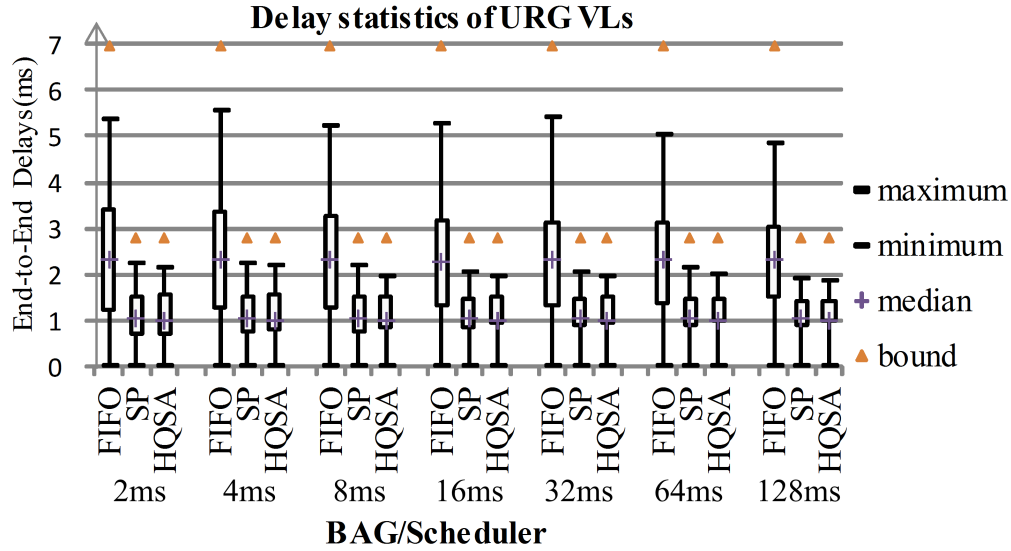


Fig. (8). End-to-End delays of flows under FIFO.

$\beta_C = C[t - \Delta]^+$ . The symbol  $\Delta$  is the transmission latency in one cable, and its value is  $\Delta = 10m / (2 \times 10^8 m/s) = 5 \times 10^{-8} s$ .

To avoid pay burst more than once, we have to analyze all elements in the route path as a whole element. For example, an urgent data flow from  $ES_1$  goes to  $ES_{10}$  through the  $SW_1$ ,  $SW_3$  and  $SW_4$ . So the service curve of the whole element is  $\beta_{E-E} = \beta_C \otimes \beta_{SW_1} \otimes \beta_C \otimes \beta_{SW_3} \otimes \beta_C \otimes \beta_{SW_4} \otimes \beta_C$ . We analyze the aggregated flows firstly, and then we split the micro data flow from the aggregated flows. We can get the delay bound of all micro data flows by referencing [14].

#### 4. SIMULATION AND RESULT

In this section, we have designed simulation platform of the AFDX network system as the model mentioned in IV and analyzed the results.

##### 1. Design of the AFDX model

We have developed a computing tool based on the toolbox of TrueTime1.5 in MATLAB. In our model, we record the delays of all types of data flows in VLs. In order to illustrate the advantages of HQSA, we realized the different AFDX model with FIFO algorithm and SP algorithm. The key policy for making the model come true is the DRR algorithm. Here we state some pseudo-code as follows:

Procedure HQSA\_DRR

WHILE (TRUE)

{

FOR i:=1 To VL\_Num

{

IF <exist frame to be transmitted in the VL>THEN

{

L:=the length of the frame to be transmitted

IF <theta >= L>THEN

Schedule the frame to the buffer

theta:=theta-L

ELSE

theta:=theta+W

}/

ELSE

theta:=0

}/

}/

End HQSA\_DRR

In this algorithm, VL\_Num is the number of VL under DRR algorithm with the same priority, theta is the counter, L is the length of data frame, and W is the amount of bandwidth allocated for each cycle. We assume the value of W as 100 bytes. The time complexity of this algorithm is  $O(VL\_Num)$ , and it is same as FIFO.

##### 2. Result

The simulation runs 1000s by different policies respectively. The statistics results of delay of the different types in VLs are shown in Fig. (8), Fig. (9), Fig. (10).

Simulation results show that, FIFO algorithm is fair for all data streams, but can't provide differentiated services based on the quality of real-time demand; SP algorithm guarantees the hard real-time data streams, but cannot guarantee the fairness and isolate the impact of malicious data flows. HQSA policy not only fulfills the requirement of real-time data, the fairness of the best effort data flow, but also isolates the impact of malicious data flow effectively.

#### CONCLUSION

Compared to FIFO algorithm adopted in current AFDX networks, we analyzed many common algorithms and proposed a novel policy, Holistic Queue Scheduling on AFDX (HQSA). We calculated the delay bound of various types of data by network calculus. Additionally, we designed a simulation platform to simulate the influence of HQSA and other

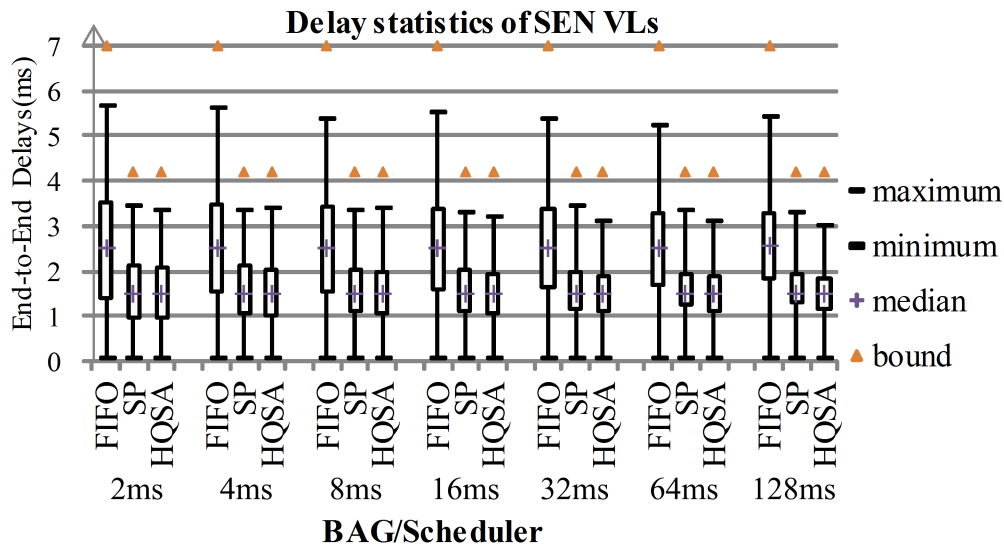


Fig. (9). End-to-End delays of flows under SP.

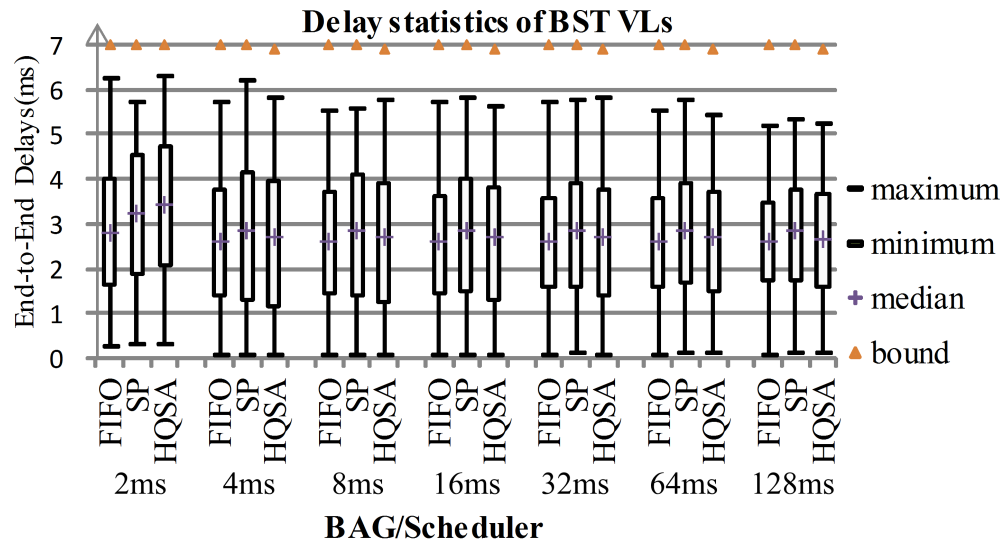


Fig. (10). End-to-End delays of flows under HQSA schedulers.

algorithms. Simulation experiments show that HQSA policy not only fulfills the demand of real-time data, the fairness of the best effort data flow, but also isolates the impact of malicious data flow effectively.

Consequently, we concluded that using HQSA policy can optimize the delay performance in AFDX.

**CONFLICT OF INTEREST**

The authors confirm that this article content has no conflicts of interest.

**ACKNOWLEDGEMENTS**

This work was supported by the Young National Natural Science Fund Project under Grant No. 61402335, The National Bureau of Statistics Research Projects under Grant No.

2012LY056, the Natural Science Fund Project in Shaanxi province under Grant No. 2012JM8031, Weinan Normal University Research Projects under Grant No.14YKS007, and Weinan Normal University Characteristic Discipline Construction Projects under Grant No.14TSXK02.

**REFERENCES**

- [1] Kai LI. The Development analysis of integrated avionics system [J]. Electronic technology and software engineering,2013,18:175.
- [2] Frazier H. The 802.3 z gigabit ethernet standard[J]. Network, IEEE, 1998, 12(3): 6-7.
- [3] ARINC. Specification 664: Aircraft data network, part7-deterministic network[S].2005.
- [4] Charara H, Scharbarg J, Ermont J, et al. Methods for bounding end-to-end delays on an AFDX network[C]. Real-Time Systems, 2006. 18th Euromicro Conference on. IEEE, 2006: 10 pp.193-202.
- [5] Cruz R. L. A calculus for network delay. I. Network elements in isolation[J]. Information Theory, IEEE Transactions on, 1991, 37(1): 114-131.

- [6] Cruz R. L. A calculus for network delay. 2. Network analysis[J]. Information Theory, IEEE Transactions on, 1991, 37(1): 132-141.
- [7] Scharbag J, Ridouard F, Fraboul C. A probabilistic analysis of end-to-end delays on an AFDX avionic network[J]. Industrial Informatics, IEEE Transactions on, 2009, 5(1): 38-49.
- [8] Ridouard F, Scharbag J, Fraboul C. Probabilistic upper bounds for heterogeneous flows using a static priority queueing on an AFDX network[C]. Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on. IEEE, 2008: 1220-1227.
- [9] CHEN Xin,ZHOU Yong-jun,JIANG Wen-bao,WAN Jian-xiong.Performance Analysis of AFDX Protocol and Scheduling Algorithm [J].ACTA Electronica Sinica,2009,37(5):1000-1005.
- [10] Dong H, Lin Y, Zhang Y, et al. Using Static Priority Queueing to optimize the Avionics Full Duplex Switched Ethernet[C]. Natural Computation (ICNC), 2013 Ninth International Conference on. IEEE, 2013: 1610-1616.
- [11] Hua Y, Liu X. Scheduling heterogeneous flows with delay-aware deduplication for avionics applications[J]. Parallel and Distributed Systems, IEEE Transactions on, 2012, 23(9): 1790-1802.
- [12] Mifdaoui A, Frances F, Fraboul C. Full duplex switched ethernet for next generation "1553B"-based applications[C]. Real Time and Embedded Technology and Applications Symposium, 2007. RTAS'07. 13th IEEE. IEEE, 2007: 45-56.
- [13] Fan Baohua.Network Performance Analysis Model Research Based on Network Calculus[D].National University of Defense Technology,2009.
- [14] Shreedhar M, Varghese G. Efficient fair queuing using deficit round-robin[J]. Networking, IEEE/ACM Transactions on, 1996, 4(3): 375-385.
- [15] ZHANG Qi-zhi, ZHANG Bin, ZHANG Wei-dong. The computing of the maximum delay in switched industrial Ethernet Based on network calculus [J]. Control and Decision,2005,20(1):117-120.

---

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Liu Jing; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.