Impact of Source Scheduling on End-to-end Latencies in a QoS-aware Avionics Network

Oana Hotescu, Katia Jaffrès-Runser, Jean-Luc Scharbarg, Christian Fraboul
Institut de Recherche en Informatique de Toulouse
Université de Toulouse, INPT
Toulouse, France
{oana.hotescu,kjr,jean-luc.scharbarg,christian.fraboul}@enseeiht.fr

ABSTRACT

This paper investigates the impact of a novel source scheduling framework tailored to introduce Quality of Service (QoS) in legacy embedded avionics networks. The protocol of interest is the Avionics Full Duplex Switched Ethernet (AFDX) deployed in modern civilian aircrafts to handle the transmission of time-critical avionics flows. Our aim is to introduce soft real-time flows on such networks in order to carry video flows. Multiplexing avionics real-time flows with video flows is tackled by introducing table scheduling at the end systems (i.e. transmitters). We propose herein a solution that preserves the worst-case end-to-end delay of avionics flows, and show how their scheduling in the table impacts the jitter of the video flows. Thorough simulations on a A350-like network architecture underline that table schedules mostly impact the jitter of video flows at the source and have limited impact on the jitter introduced by the network transfer.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

AFDX; Video transmission; Table scheduling; End-to-end delay;

ACM Reference Format:
Oana Hotescu, Katia Jaffrès-Runser, Jean-Luc Scharbarg, Christian Fraboul. 2019. Impact of Source Scheduling on End-to-end Latencies in a QoS-aware Avionics Network. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3297280.3297553

1 INTRODUCTION

Avionics Full-Duplex Switched Ethernet (AFDX)[1] is adopted as a backbone network by most of the civilian aircraft manufacturers due to its high throughput provision with reduced implementation costs. The AFDX is certified to ensure that avionics messages arrive within a bounded time. Safe upper bound on the end-to-end delay of frames can be computed using network calculus [3] or trajectory approach[2]. However, this bound is pessimistic and leads to an over-provisioning of the network bandwidth. Thus, the avionics industry envisions to use the available AFDX bandwidth for the transmission of additional flows of lower criticality such as video, audio, or service data flows, currently carried by dedicated communication networks. Nevertheless, the transmission of additional flows cannot compromise the timely transmission of avionics flows. Jitter of critical flows at source level and end-to-end latency have to respect the upper bounds imposed by the certification body. This negative impact can be mitigated thanks to QoS policies, implemented both at switch and at end system levels. For instance, at switch egress ports, Static Priority Queuing (SPQ) scheduling policy is currently deployed with 2 priority levels and a worst case delay analysis has been provided for this policy [4]. In this paper, we focus on video streams originating from cameras located on the aircraft fuselage and wings. Their jitter and end-to-end latency have to be minimized in order to provide a good quality of experience to the pilot and crew members. Frame scheduling in source end systems introduces offsets between flows. It determines the jitter of flows at source level and has an impact on flow end-to-end latency. It has been shown that worst case end-to-end delays can be improved by considering the offsets between flows [5]. A solution based on table scheduling where slots are statically assigned to flows is presented in [7]. This solution is promising as it permits to schedule flows with different criticality levels and mitigate avionics flows jitter.

In this paper, we focus on the table scheduling as a potential method to schedule avionics and additional flows on AFDX at source level. Our goal is to determine how different configurations of the table schedule impact the end-to-end delay of video flows. More specifically, we evaluate the impact of scheduling policies on the jitter of video flows at end system output and after the network traversal by a set of extensive simulations of naive and better engineered tables. Depending on their impact, we can decide in a later stage on the best way to built optimal QoS-enabled policies at end system or at network level, or at both levels at the same time.

2 NETWORK MODEL

The architecture is based on an AFDX network implementing quality of service to share avionics and video flows.

2.1 AFDX and ADVB

Avionics Full DupleX switched Ethernet(AFDX)[1] is a switched Ethernet-based network that has been designed for safety-critical avionics applications. The main elements composing the AFDX network are the network emitting cards (or end systems) interconnected by switches and physical links. An end system is connected to a switch by a unique physical link. Data exchange between end

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3297553

systems can occur only through virtual communication channels called Virtual Links (VL) defined as a set of unidirectional paths connecting a source to one or more destinations. Each VL is assigned a maximum bandwidth share specified by the Bandwidth Allocation Gap (BAG) and the authorized maximum frame size. The BAG represents the minimum time interval between two consecutive frames transmitted through the VL. The BAG value is given in the set of powers of 2: {1, 2, 4,. . . , 128} milliseconds. AFDX switches are composed of a set of ingress ports and a set of egress ports. Multicast VL routes are statically configured on the switches by static switching tables. For each egress port, the ARINC 666-P7 standard specifies two egress queues with different priority levels. In the existing deployed networks, only the high priority level is used by avionics frames. Therefore, the lower priority can be assigned to additional flows. At end system level, the BAG constraint is guaranteed by a traffic regulator to each VL. Frames coming from the regulators are multiplexed by the scheduler into a single flow to be sent on the output physical link. If multiple frames from different VLs arrive at scheduler input at the same time, some of them may experience queuing delay or jitter. The ARINC 664 P7-1 standard specifies that the jitter for a given VL must not exceed 500 μs to limit the impact on determinism for the whole network. Therefore, the end system scheduler needs to implement an appropriate scheduling policy to keep jitter within fixed boundaries. A possible solution based on a table scheduling has been proposed in [7]. The principle of the table scheduling is to statically assign periodic slots to each VL. The advantage of such a periodic schedule of VLs is that their jitter is mitigated at end system output. In the AFDX network, there is no global clock. Consequently, transmitting end systems are not synchronized. Thus, the start dates of computing module are unknown. AFDX network rate is of 100 Mbps, and will be of 1 Gbps soon.

Avionics Digital Video Bus (ADVB)[8] is the standard video communication protocol for avionics systems. It specifies that only *uncompressed* digital video streams can be exchanged in the avionics systems to mitigate stream reconstruction delays at the receiver. This standard is based on Fiber Channel Audio Video (FC-AV) protocol which is particularly suited for high bandwidth communications of high-resolution video streams. The delivery of video data without compression is needed by the avionics systems supporting time-critical functions such as assisted takeoff and landing, navigation, etc. Thus, neither frame losses nor additional latency introduced by compression/decompression operations are tolerated. A video is composed of a sequence of images referred to as video frames. A video frame is transmitted as a sequence of data packets called ADVB frames. The sequence of ADVB frames form an ADVB container that is specific to a single video frame. ADVB frames are labeled with an *object* field that represents the type of data carried in the frame: object 0 contains video frame header data (image information and auxiliary data), object 1 contains audio data, object 2 or 3 frames contain video data. For the transmission of an image, the first frame is object 0 type and lists information related to the image (sequence number, type of color coding, dimensions, etc.). ADVB standard specifies that, to encapsulate video data into frames, images are scanned from left to right, line by line. If the last frame's payload for a line is not complete, padding bytes are added to keep the frame size constant. The ADVB frame is very similar to the frame used for Ethernet with a maximum frame length of 1518 bytes including 1500 bytes of payload. Therefore, AFDX network is a good candidate to deliver video in accordance with ADVB.

2.2 Problem statement

The AFDX network configuration considered in this paper is defined as a set of end systems emitting avionics frames with real-time constraints through virtual links together with additional video flows. We summarize here the main constraints to be ensured for both avionics and video flows. *i*) For avionics real-time flows, the maximum allowed jitter at source end system is 500μs as specified in the ARINC 664 standard. Moreover, an upper bound on the end-to-end delay needs to be guaranteed. *ii*) Video frames are emitted periodically, according to the ADVB standard. To provide a sufficient quality of video at destination, it is important to ensure low jitter and delay. In Figure 1, an AFDX network containing a small subset of the end systems and 7 switches is depicted as well as 5 VLs generated by end system *ES_1* and a video flow crossing switch *SW_1*. Each network component introduces a delay for each frame. At source end system, frames arriving simultaneously at scheduler input may experience queuing delay or jitter. This delay introduced by the scheduler is referred in this paper as the source delay. The delay required by a frame to cross all the physical links and switches to arrive at destination is defined as network crossing delay. The end-to-end delay of video frames may be different depending on the scheduling policy implemented at source or switch level. It has been shown in [6] that a FIFO scheduling cannot guarantee jitter constraints for avionics flows at the end system level. Conversely, table scheduling as introduced in [7] is able to bound this jitter. The rest of the paper considers table scheduling in end systems.

3 TABLE SCHEDULING POLICIES FOR AFDX

This section introduces our implementation of table scheduling which has been tailored to carry VLs together with video flows in AFDX. Two possible VL allocation schemes are introduced as well.

In order to reduce the impact of video frames on VL frames inside the network, SPQ policy is considered at the switch level with 2 levels of priority. High priority is assigned to avionics hard real-time VLs and low priority to the video flows.

3.1 Table scheduling

The definition of a table schedule is motivated by the maximum jitter requirement of VL flows specified in the ARINC 664 P7-1

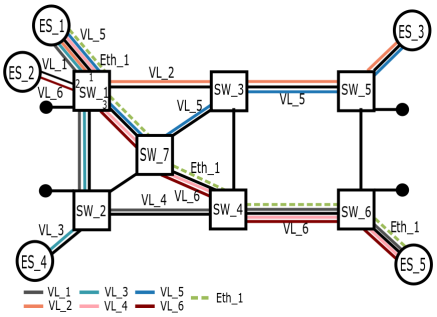

Figure 1: AFDX configuration with ES_1 source of 5 VLs

standard. Any AFDX end system must ensure that the jitter of VLs does not exceed 500μs. With proper scheduling operations, it is possible to mitigate the jitter of real-time flows at the end system. Our table is built during the network design stage and is played cyclically at execution time. The table is modeled as a 2D matrix of time slots that can be allocated to the demanding flows of the end system. The table duration is 128ms which corresponds to the largest possible hyper-period of all possible BAG values of VLs. The table is composed of 128 lines of 1ms, each line being divided into 64 slots of 15.625μs. According to this time division, the table can be represented as a 2D matrix of 128 lines $\times$ 64 columns. It is convenient with this table to assign transmission opportunities to VLs every BAG units of time. For instance, if a VL is assigned to a single slot, it receives a transmission opportunity every 128ms. If it is assigned a full column, it can transmit every millisecond. Such a schedule ensures that the frames of a VL leave the end system at dates multiple of its BAG, completely mitigating the jitter of the VL at emission. As such, a VL frame can only experience additional delay at egress ports of switches due to the interference with other VL frames or to the non-preemption delay induced by lower priority video frames. The remaining slots in the table can be allocated to other types of flows.

3.2 Slot allocation policies

We define next two simple slot allocation policies that produce drastically different end-to-end delay statistics for video flows. In both allocation schemes, we provision a full column among the 64 of the table for each VL. Allocating a column to a VL is intended to minimize the queuing delay of frames before transmission and ensures that all its frames wait for at most one millisecond.

The Naive Allocation (NA). This naive table configuration assigns the first K columns of the table to the K VLs. The rest of slots is free for carrying additional flows. This configuration is valid for VLs as they will not suffer from any jitter. However, our intuition tells us that this arrangement does not benefit video flows as they will be subject to jitter equal to at least the number of VLs times the slot duration. A slightly improved solution consists in assigning a column every two columns to the VLs. Thus, there is a gap of one slot every two slots for additional flows leaving them the possibility to receive slots more frequently than before. Further analysis is needed to establish if this delay is small enough to keep the jitter of additional flows at minimum.

The Uniformly Distributed Allocation (UDA). Another possible configuration consists in spreading the column assigned to VLs on the complete table duration. As a result, the gap between assigned columns depends on the number of VLs and is represented as the total number of columns divided by the number of VLs. The choice of the columns to be assigned to VLs is still arbitrary, but the gap between columns is considered. For example, the 5 VLs of ES_1 in Figure 1 can be assigned every 12 columns approximately.

4 END-TO-END DELAY STUDY

The goal of this paper is to highlight the impact of table scheduling policies on the distribution of source, network and end-to-end delay of video flows. The distribution of avionics flows is not further discussed here as it has been shown in previous works [7] that little impact is observed. Moreover, the analytical worst-case delay bounds of VLs still holds in our setting.

4.1 Simulation settings

Following results were produced by an in-house AFDX network simulator whose core relies on the OMNeT++ discrete event framework [9]. Our simulator models the main parameters (routing of virtual links, queuing policies, switching latency, etc.) of an AFDX network configuration similar to the one deployed on the Airbus A350 airplane. 1106 avionics VLs are carried in a network similar to the one depicted in Figure 1. In addition to this basic configuration, one of the end systems supports the transmission an uncompressed video flow. The network data rate is set to 1 Gbps and the switching delay to 2μs. The video flow and avionic VLs follow the same path and cross 3 switches. Thus, for a maximum frame length, the minimum end-to-end delay expected corresponds to 54.576 μs. The selected end system is source of a given number of VLs characterized by BAG values from 16 ms to 128 ms and a maximum frame length of 971 bytes. VL frames are generated with a frame generation period of at least one BAG duration. Video flows follow the ADVB specification. We retain for our simulations the XGA video display format with 3-byte color and a resolution of 1024 pixels $\times$ 768 lines per image. The length of video frames is fixed to 1518 bytes. A video displaying 24 frames per second has to receive 55320 Ethernet frames per second with a bandwidth requirement of 640.685 Mbps. Thus, the period of video frame generation equals 18.076μs for XGA format. Each end system plays NA and UDA table schedule policies described in section 3. Even though global synchronization is not authorized in the AFDX standard, we introduce relative offsets in our simulations to represent the possible shift between the power on dates of end systems. To capture the delay distribution of a wide range of possible scenarios, Monte-Carlo simulations are performed here. The relative offsets of end systems take a new randomly chosen value in the $[0..128ms]$ interval at each new simulation. The simulation duration has been set to $384ms$, corresponding to 3×128 ms. It is equivalent to 2 times the duration of the table schedule (128 ms) in addition to the maximum offset value of 128ms. This duration captures the complete delay distribution. We consider the distribution delay for the first 14162 frames arrived at their destination for XGA display format.

4.2 Results

Results are given for the case where i) the relative offsets are null and ii)random relative offsets are introduced. Our simulations measure the impact of different table schedules on the delay distribution of video flows. Results are shown for the case where the end system under investigation is the source of 5 VLs and of one XGA flow.

4.2.1 Null offset case study. All relative offsets in the network are equal to zero, which corresponds to a perfectly synchronized network. This setting is rarely experienced in reality. In this case, as many VLs have similar BAGs, a larger network delay is expected as congestion in the switch egress ports is the norm. Figure 2 represents the Complementary Cumulative Distribution Function (CCDF) of the source, network and end-to-end delays experienced by the XGA video flow for both NA and UDA schedules. The largest source delay experienced with the Naive Allocation is around 80

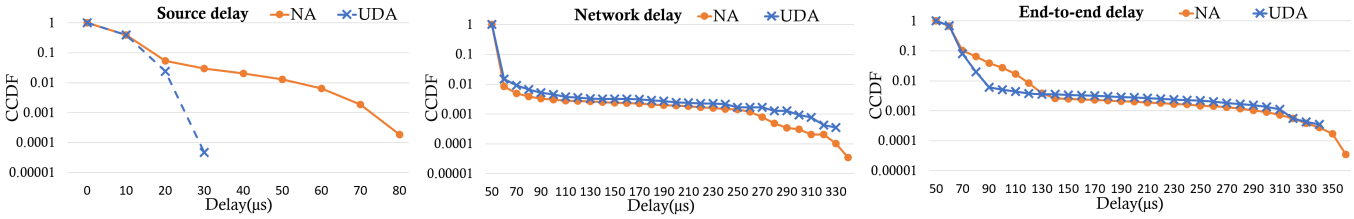

Figure 2: CCDF of the source delay (left), network delay (center) and end-to-end delay (right) for NA and UDA table schedules. The tested scenario assumes no offset exists in-between the end systems (offset 0). Statistics are derived from the first 14162 video frames arrived. This scenario considers 5 VLs and 1 XGA video flow.

μs. This delay is rarely seen and most of the messages are in the $[0, 10\mu s]$ interval, with a median value of $10\mu s$. The UDA triggers a source delay of about $30\mu s$, offering improvement to the NA scheme. It can be noticed as well that the maximum network crossing delay is 350 μs and that the peak of video frames is reached at 54.576 μs which corresponds to the shortest network crossing delay possible experienced when the frame never waits in switch queues. With UDA, the maximum network crossing delay is similar and of 338 μs. The peak of video frames is reached at the minimum delay value of 54.576 μs. Thus, table schedules seems to have little impact on the network crossing delay statistics.

4.2.2 Random offsets case study. Monte-Carlo simulations have been performed here to capture the impact of various offset combinations on the source, network and end-to-end delays. 100 simulation runs are necessary to offer a complete delays distribution. Figure 3 represents the CCDF of the network and end-to-end delays experienced by the XGA video flow for both NA and UDA table schedules. The source delay is the one in Figure 2, as slot allocations NA and UDA are the same as for the null offsets case. Similarly to the null offset scenario, network delay is in the same order of magnitude whether NA or UDA are implemented. The maximum network delay for NA is $80\mu s$ and 79.797 μs for UDA. Most of the frames experience the minimum network traversal time of 54.576 μs for both cases. Note that only very few video frames experience

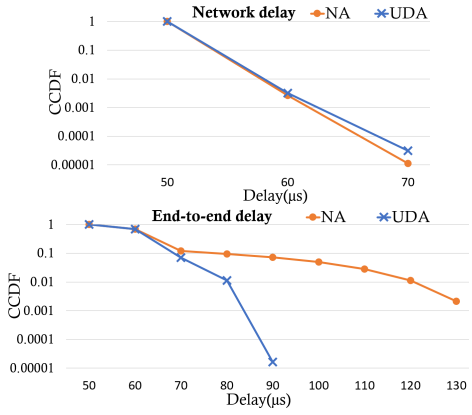

Figure 3: CCDF of the network delay (up) and end-to-end delay (down) for NA and UDA table schedules. 100 simulation runs are repeated for random relative offset values. Statistics are derived from the first 14162 video frames arrived.

a bad network traversal time. 99.9% of video frames reach their destination within $80\mu s$ with the UDA scheme. This represents an improvement of 60% compared to the NA scheme. The following insight is gained from our study: *i*) Table scheduling is well suited to introduce QoS flows in the AFDX network ; *ii*) Table schedules mostly influence the source delay. The little impact on network delay is explained by the reduced number of video flows transiting in the network. *iii*) Building good schedules is clearly beneficial to the overall end-to-end delay. Thus, future work will concentrate on defining optimal table schedules at the end systems.

5 CONCLUSION

We have shown that scheduling both critical and video flows at the end system can be really beneficial for video flows. Simulation results show that, on average, the source delay dominates the end-to-end delay. This source delay of video flows is strongly influenced by the slot allocation policy. It follows that the table schedule has to be wisely chosen. As for the network crossing delay, since end systems are not synchronized in AFDX, the delay of video flows is not significantly impacted by congestion of avionics frames at switch level. Future work will first focus on how to derive optimal table schedules to ensure bounded delays for avionic flows and a minimal jitter for the additional video flows. Next, we will investigate the use of advanced scheduling policies at switch egress ports. Lastly, we plan to discuss the transport of compressed video flows since it offers an opportunity to carry several flows.

REFERENCES

[1] Aeronautical Radio Inc. ARINC specification 664 P7-1, *Aircraft Data Network, Part 7: Avionics Full Duplex Switched Ethernet (AFDX) Network*, 2009.
[2] H. Bauer, J.-L. Scharbarg, and C. Fraboul, *Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network*, IEEE Emerging Technologies and Factory Automation (ETFA), 2009.
[3] H. Charara, J. L. Scharbarg, J. Ermont and C. Fraboul. *Methods for bounding end-to-end delays on an AFDX network*, Proceedings of the 18th Euromicro Conference on real time Systems (ECRTS'06), Dresden, 2006.
[4] H. Bauer, J.-L. Scharbarg and C. Fraboul. *Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources*, real time systems 48.1, pp 101-133, 2012.
[5] X. Li, J.-L. Scharbarg and C. Fraboul, *Improving end-to-end delay upper bounds on an AFDX network by integrating offsets in worst-case analysis*, IEEE Emerging Technologies and Factory Automation (ETFA), 2010 .
[6] M. Boyer, L. Santinelli, N. Navet, J. Migge, and M. Fumey, *Integrating end-system frame scheduling for more accurate AFDX timing analysis*, Embedded Real-Time Software and Systems (ERTS 2014), Toulouse, France, 2014.
[7] O. Hotescu, K. Jaffrès-Runser, J.-L. Scharbarg and C. Fraboul, *Towards Quality of Service Provision with Avionics Full Duplex Switching*, ECRTS 2017 Work-in-Progress Session, pp 19-21, 2017.
[8] ARINC specification 818-2 *Avionics Digital Video Bus (ADVB) High Data Rate*, 2013.
[9] A. Varga and R. Hornig. *An overview of the OMNeT++ simulation environment.* In Proceedings of Simutools '08. ICST, Brussels, Belgium, 2008.