

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

**DYNAMIC PRIORITY ASSIGNMENT FOR IMPROVING
SCHEDULABILITY AND MIXED CRITICALITY IN
AVIONICS NETWORKS**

Metin YENİAYDIN

MASTER OF SCIENCE THESIS
Department of Avionics Engineering
Program of Avionics Engineering

Supervisor
Prof. Mehmet Serdar Ufuk TÜRELİ

Co-supervisor
Dr. İbrahim HÖKELEK

February, 2022

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

**DYNAMIC PRIORITY ASSIGNMENT FOR IMPROVING
SCHEDULABILITY AND MIXED CRITICALITY IN AVIONICS
NETWORKS**

A thesis submitted by Metin YENİAYDIN in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 21.02.2022 in Department of Avionics Engineering, Program of Avionics Engineering.

Prof. Mehmet Serdar Ufuk TÜRELİ
Yildiz Technical University
Supervisor

Dr. İbrahim HÖKELEK
TÜBİTAK BİLGEM
Co-supervisor

Approved By the Examining Committee

Prof. Mehmet Serdar Ufuk TÜRELİ, Supervisor
Yildiz Technical University

Assoc. Prof. Hacı İLHAN, Member
Yildiz Technical University

Asst. Prof. Selçuk CEVHER, Member
Karadeniz Technical University

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Dynamic Priority Assignment for Improving Schedulability and Mixed Criticality in Avionics Networks supervised by my supervisor, Prof. Mehmet Serdar Ufuk TÜRELİ. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Metin YENİAYDIN

Signature

Dedicated to my family



ACKNOWLEDGEMENTS

I would like to first express my gratitude to my advisor Prof. Mehmet Serdar Ufuk TRELİ, for his guidance and support. I would also like to express my sincere appreciation to my co-advisor, Dr. İbrahim HKELEK, who always does his best for us with all sincerity, always encourages and supports us, and always behave more than a supervisor.

I would like to express my gratitude to Dr. Muhammet Selim DEMİR, who made outstanding contributions on this thesis and guided me with his valuable perspective. I would like to thank mer Faruk GEMİCİ and Sinem LERİ, who significantly broadened my horizon with their fruitful feedbacks. I also would like to thank my colleagues at TBİTAK BİLGEM for their valuable feedback and the daily discussions.

The warmest part of my gratitude belongs to my family. I would like to thank my mother, Şkran, and my father, Hasan, for their endless love and support that I will never be able to reciprocate. I would like to express my special thanks to my brothers, Burak and Muhammed Emin, and my sister, Rmeysa, for their unconditional supports and sincere encouragement.

Metin YENİAYDIN

TABLE OF CONTENTS

LIST OF SYMBOLS	vii
LIST OF ABBREVIATIONS	ix
LIST OF FIGURES	xi
LIST OF TABLES	xiii
ABSTRACT	xiv
ÖZET	xvi
1 INTRODUCTION	1
1.1 Literature Review	3
1.2 Objective of the Thesis	7
1.3 Hypothesis	8
1.4 Organisation of the Thesis	9
2 BACKGROUND	10
2.1 Time Sensitive Network	10
2.2 ARINC 664	14
2.2.1 Virtual Link	15
2.2.2 End System	16
2.2.3 Switch	18
2.3 Network Calculus	19
2.3.1 Network Calculus Analysis for Switch	19
2.3.2 Network Calculus Analysis for End System	22
3 SYSTEM MODEL	23
3.1 Single-Hop Model	23
3.2 Multi-Hop Model	24
3.3 Dynamic Priority Assignment Algorithm	25
3.4 Network Calculus Delay Analysis	27
3.4.1 Network Calculus for Single-Hop Model	27

3.4.2 Network Calculus for Multi-Hop Model	30
4 PERFORMANCE EVALUATION	32
4.1 Performance Evaluation for Single-Hop Model	32
4.1.1 Numerical Experiments for Single-Hop Model	32
4.1.2 Simulation Analysis for Single-Hop Model	36
4.2 Performance Evaluation for Multi-Hop Model	42
5 RESULTS AND DISCUSSION	48
REFERENCES	49
PUBLICATIONS FROM THE THESIS	54



LIST OF SYMBOLS

$\alpha_n(t)$	Arrival Curve of n^{th} Virtual Link
r_n	Arrival Rate of n^{th} Virtual Link
$\alpha_{HP}(t)$	Arrival Rate of the HP Flows
$\alpha_{LP}(t)$	Arrival Rate of the LP Flows
BAG_n	Bandwidth Allocation Gap of n^{th} Virtual Link
I_{idle}	Credit Decreases Rate
I_{send}	Credit Increases Rate
N	Initial HP Flow Count
M	Initial LP Flow Count
$\Delta_{send,0}^{max}$	Initial Maximum Sending Window Duration
L_M	Maximum Credit Threshold
$D_{max,Thr}$	Maximum Delay Threshold
s_n	Maximum Frame Size of n^{th} Virtual Link
s_{HP}	Maximum Frame Size of the HP Flows
s_{LP}	Maximum Frame Size of the LP Flows
Δ_{send}^{max}	Maximum Sending Window Duration
Υ_{HP}^{bls}	Maximum Service Curve of the BLS for HP Class
$D_{max,HP}$	Maximum Worst-case Delay of HP
$D_{max,LP}$	Maximum Worst-case Delay of LP
L_R	Minimum Credit Threshold
Δ_{idle}^{max}	Minimum Idle Window Duration
Δ_{send}^{min}	Minimum Sending Window Duration
$\beta_{HP}^{bls}(t)$	Minimum Service Curve of the BLS for HP Class

$\otimes$	Min-plus Convolution Operator
$\oslash$	Min-plus Deconvolution Operator
$\alpha_n^*(t)$	Modified Arrival Curve
$\alpha_{HP}^{*,bls}(t)$	Modified Arrival Curve after Crossing BLS
$\beta_{HP}^{mux}(t)$	Multiplexed Service Curve of HP Class
$\beta_{LP}^{mux}(t)$	Multiplexed Service Curve of LP Class
D_{HP}^{mux}	Multiplexed Worst-case Delay of HP Class
D_{LP}^{mux}	Multiplexed Worst-case Delay of LP Class
f_n	n^{th} Flow
VL_n	n^{th} Virtual Link
$\Delta\tau_{SCT}$	Period of the Maximum Service Curve
$p(i)$	Priority Level of i^{th} Flow
R_n	Remaining Service Capacity
$\beta_j^{sp}(t)$	Residual Strict Service Curve
R	Service Capacity
$\beta_n(t)$	Service Curve of n^{th} Virtual Link
β_{LP}^{bls}	Service Curve of the BLS for LP Class
$\beta_{HP,0}^{sp}$	Service Curve of the HP Class when it has Higher Priority than LP Class
$\beta_{HP,2}^{sp}$	Service Curve of the HP Class when it has Lower Priority than LP Class
β_{LP}^{sp}	Service Curve of the LP Class
T	Technological Latency
T_n	Technological Latency Parameter
$D_j^{sp}(t)$	Worst-case Delay for Strict Priority Scheduler
$D_{SW}^{f_n}$	Worst-case Delay of n^{th} Flow
UR_{HP}	Utilization Rate of the HP Class
UR_{LP}	Utilization Rate of the LP Class

LIST OF ABBREVIATIONS

ARINC	Aeronautical Radio, Incorporated
AVB	Audio Video Bridging
Avg.	Average
BAG	Bandwidth Allocation Gap
BE	Best-Effort
BLS	Burst Limiting Shaper
CBS	Credit-Based Shaper
DPA	Dynamic Priority Assignment
DRR	Deficit Round Robin
DVD	Digital Versatile Disc
ES	End System
FIFO	First-In First-Out
FPA	Fixed Priority Assignment
GA	Genetic Algorithm
GCL	Gate Control List
Gbps	Gigabit per Second
HP	High Priority
IEEE	Institute of Electrical and Electronics Engineers
IMA	Integrated Modular Avionics
IS-IS	Intermediate System to Intermediate System
Kbps	Kilobit per Second
LP	Low Priority
MAC	Media Access Control

MFS	Maximum Frame Size
MIL-STD	Military Standard
MPA	Modular Performance Analysis
Mbps	Megabit per Second
OMNeT++	Objective Modular Network Test-bed in C++
OSI	Open Systems Interconnection
PID	Proportional Integral Derivative
PROFIBUS	Process Field Bus
PROFINET	Process Field Net
PS	Peristaltic Shaper
PTP	Precision Time Protocol
QoS	Quality of Service
RC	Rate Constrained
SCT	Safety-Critical Traffic
SP	Strict Priority
SW	Switch
TAS	Time-Aware Shaper
TSN	Time Sensitive Network
TT	Time-Triggered
TTEthernet	Time-Triggered Ethernet
UBS	Urgency Based Scheduler
VL	Virtual Link
WbA	Window-based Approach
WC	Worst-Case
XML	Extensible Markup Language

LIST OF FIGURES

Figure 2.1	Time aware shaper gating mechanism [41]	11
Figure 2.2	Operation of the burst limiting shaper [39]	13
Figure 2.3	ARINC 664 cascade star topology	14
Figure 2.4	Transmitting side of the ARINC 664 end system [53]	16
Figure 2.5	Regulating unshaped frames	16
Figure 2.6	Jitter [53]	17
Figure 2.7	Receiving side of the ARINC 664 end system [19]	17
Figure 2.8	ARINC 664 switch block diagram	18
Figure 2.9	Calculating the worst-case delay	20
Figure 3.1	Single-hop system model	23
Figure 3.2	Multi-hop system model	24
Figure 3.3	Algorithm flow chart	26
Figure 4.1	The delay performance of FPA and DPA as UR_{HP} varies	33
Figure 4.2	The delay performance of FPA and DPA as UR_{LP} varies	34
Figure 4.3	The delay performance of BLS-FPA and BLS-DPA as UR_{HP} varies	35
Figure 4.4	The delay performance of BLS-FPA and BLS-DPA as UR_{LP} varies	36
Figure 4.5	Simulation environment for single-hop system model	37
Figure 4.6	End system simulation model inner structure [61]	38
Figure 4.7	Switch module	38
Figure 4.8	The switch port module	39
Figure 4.9	Switch fabric module	39
Figure 4.10	The comparison of the worst-case and simulation delay of HP class as UR_{HP} varies (single-hop scenario)	40
Figure 4.11	The comparison of the worst-case and simulation delay of LP class as UR_{HP} varies (single-hop scenario)	41
Figure 4.12	The comparison of the worst-case and simulation delay of HP class as UR_{LP} varies (single-hop scenario)	41
Figure 4.13	The comparison of the worst-case and simulation delay of LP class as UR_{LP} varies (single-hop scenario)	42
Figure 4.14	Multi-hop scenario	43
Figure 4.15	OMNeT++ implementation of multi-hop system model	43

Figure 4.16 The comparison of the worst-case and simulation delay of HP class as UR_{HP} varies (multi-hop scenario)	44
Figure 4.17 The comparison of the worst-case and simulation delay of LP class as UR_{HP} varies (multi-hop scenario)	45
Figure 4.18 The comparison of the worst-case and simulation delay of HP class as UR_{LP} varies (multi-hop scenario)	46
Figure 4.19 The comparison of the worst-case and simulation delay of LP class as UR_{LP} varies (multi-hop scenario)	46



LIST OF TABLES

Table 2.1	IEEE TSN standards [17]	10
Table 4.1	Number of flows generated by each end system group	37



Dynamic Priority Assignment for Improving Schedulability and Mixed Criticality in Avionics Networks

Metin YENİAYDIN

Department of Avionics Engineering

Master of Science Thesis

Supervisor: Prof. Mehmet Serdar Ufuk TÜRELİ

Co-supervisor: Dr. İbrahim HÖKELEK

ARINC 664, which is a heavily used protocol for modern avionics networks, is preferred due to its simplicity, although its mixed-criticality support is limited. Time Triggered Ethernet (TTEthernet) and IEEE Time Sensitive Networking (TSN), which utilize a time-synchronized schedule, are more suitable for supporting mixed-criticality applications; however, both require a fault-tolerant time synchronization that makes the certification process more challenging. In this paper, we propose a novel dynamic priority assignment (DPA) concept together with the burst limiting shaper (BLS) from the IEEE TSN standard to enhance the schedulability and the mixed-criticality support of ARINC 664. The decision of flow re-assignment to a new priority class is done by calculating the high priority (HP) and low priority (LP) class worst-case delays using the network calculus framework. Two different case studies are utilized, namely single-hop and multi-hop, to verify the effectiveness of the proposed algorithm and to evaluate it on realistic network topology, respectively. The performance of the proposed algorithm is demonstrated by both theoretical calculations and simulations. The theoretical results show that the class utilization rates can be significantly increased by using the DPA concept with and without the BLS. For all experiments, the delays obtained from the simulations are consistently below the theoretical upper bound delay values and the graphs obtained from the simulations and the theoretical delay results show similar characteristics. These two outputs demonstrate that the simulation results confirm the theoretical results. As

a result, the DPA can improve the schedulability and mixed-criticality of ARINC 664 without using any time synchronization mechanism.

Keywords: ARINC 664, deterministic network, mixed-criticality, schedulability, IEEE TSN-BLS



Aviyonik Ağlarda Programlanabilirliği ve Karma Kritikliği Geliştirmek için Dinamik Öncelik Ataması

Metin YENİAYDIN

Aviyonik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Prof. Mehmet Serdar Ufuk TÜRELİ
Eş-Danışman: Dr. İbrahim HÖKELEK

Modern aviyonik ağlar için yoğun olarak kullanılan bir protokol olan ARINC 664, karma kritiklik desteği sınırlı olmasına rağmen karmaşıklık seviyesinin düşük olması nedeniyle tercih edilmektedir. Zamansal olarak senkronize edilmiş zaman planı uygulayıcısı kullanan Zaman Tetiklemeli Ethernet (TTEthernet) ve IEEE Zamana Duyarlı Ağ (TSN), farklı kritiklik seviyesinde olan uygulamaları desteklemek için daha uygundur; ancak her ikisi de sertifikasyon sürecini daha zorlu hale getiren hataya dayanıklı bir zaman senkronizasyonu gerektirir. Bu tezde, ARINC 664'ün programlanabilirliğini ve karma kritiklik desteğini geliştirmek için IEEE TSN standartlarından biri olan ani trafik patlamasını sınırlayıcı (BLS) ile birlikte yeni bir dinamik öncelik ataması (DPA) konsepti öneriyoruz. Akışın yeni bir öncelik sınıfına yeniden atanması kararı, teorik ağ hesabı yaklaşımı kullanılarak yüksek öncelikli (HP) ve düşük öncelikli (LP) sınıfların en kötü durum gecikmeleri hesaplanarak alınır. Önerilen algoritmanın etkinliğini doğrulamak ve gerçekçi bir ağ topolojisi üzerinde değerlendirmek için sırasıyla tek sekmeli ve çoklu sekmeli olmak üzere iki farklı durum çalışması kullanılmıştır. Önerilen algoritmanın performansı ise hem teorik hesaplamalar hem de gerçekleştirilen simülasyonlar ile gösterilmiştir. Teorik hesaplamalar, DPA konseptinin BLS kullanımından bağımsız olarak sınıf kullanım oranlarını önemli ölçüde arttırabileceğini göstermektedir. Tüm deneyler için simülasyondan elde edilen gecikmeler sürekli olarak teorik üst sınır gecikme değerlerinin altındadır ve simülasyondan elde edilen grafikler ve teorik gecikme sonuçları benzer özellikler göstermektedir. Bu iki çıktı, simülasyon sonuçlarının

teorik sonuçları doğruladığını göstermektedir. Sonuç olarak, DPA, herhangi bir zaman senkronizasyon mekanizması kullanmadan ARINC 664'ün programlanabilirliğini ve karma kritikliğini iyileştirebilir.

Anahtar Kelimeler: ARINC 664, deterministik ağ, karma-kritiklik, programlanabilirlik, IEEE TSN-BLS



1

INTRODUCTION

Next-generation aircraft have enormous data exchange requirements among avionics applications. Since these applications carry out critical tasks, they need to work on a deterministic system. Determinism can be defined as the ability to produce a predictable result usually based on previous operations [1]. Therefore, real-time computing and deterministic data transmission are two crucial concepts for the deterministic system.

Deterministic data communication has been traditionally provided with the avionics communication standard such as MIL-STD 1553 and ARINC 429 [2] [3] [4]. Although these standards bring too many advantages, such as good quality of service (QoS) and easy testing, there are some limitations, such as low bandwidth and increasing weight in parallel with the number of connections. To explain, in ARINC 429 protocol, each endpoint that is required to communicate is connected to each other individually [5] [6]. Therefore, each required connection needs extra cabling, which significantly increases the aircraft's weight. However, together with advances in avionics technologies such as the Integrated Modular Avionics (IMA) concept, the interlinking multitude of endpoints has become an important issue [7]. As a result, the popularity of ARINC 429 decreases for recently developed avionics systems. Modern avionics systems also need high bandwidth communication architecture that allows massive data transmission. However, the bandwidth is a significant limitation for ARINC 429 and MIL-STD 1553, whose maximum values are 100 Kbps and 1 Mbps, respectively [5]. Furthermore, considering the bandwidth requirements that are expected to increase continuously, the need for a different technology infrastructure that can support higher data communication speeds such as Ethernet becomes evident [8].

The standard Ethernet guarantees neither packet delivery nor bounded latency and jitter (packet delay variation). This non-deterministic behavior is not suitable for the avionics systems [9]. Therefore, Ethernet-based deterministic network technologies such as Time-Triggered Ethernet (TTEthernet) [10], IEEE Time Sensitive

Network (TSN) [11], and ARINC 664 [12] standards have been proposed. These technologies include additional algorithms to provide determinism. Furthermore, IMA architecture requires a mixed-criticality concept that allows transmission of different time-critical applications' traffic over a single physical network [13] [14]. Ethernet-based deterministic network technologies provide a suitable environment for mixed-criticality support [15].

TTEthernet and IEEE TSN implement time-triggered communication mechanisms that establish and maintain a global time through time synchronization. The synchronization mechanism helps to establish temporal partitioning and ensures isolation of the synchronous time-critical data flows from other traffic [16]. Time-triggered communication is suitable for periodic command and control tasks or synchronous data delivery with low latency and jitter requirements. TTEthernet offers three prioritization levels, time-triggered (TT), rate constrained (RC), and best-effort (BE), by allowing mixed-criticality applications to share a single physical network. Similarly, IEEE TSN also supports up to eight distinct traffic classes. Both protocols support mixed-criticality by compromising increasing complexity with time synchronization. TTEthernet has been specifically developed for the avionics systems while the IEEE TSN standards are proposed to unify different industrial standards that need time-synchronized low latency streaming services [17]. In addition to its extensive usage in automotive and industrial control applications, there is a growing interest to utilize IEEE TSN standards for avionics networks [18]; however, to the best of our knowledge, there is no certifiable IEEE TSN solution up to this date for avionics applications. For both TTEthernet and IEEE TSN, the performance will be significantly degraded if time synchronization is lost due to hardware, software, and network failures. Therefore, ARINC 664 protocol, which does not need time-synchronization, comes into prominence.

ARINC 664 is a profiled network that is a heavily used protocol for modern avionics networks due to its simplicity. In this technology, the flows are shaped based on leaky-bucket algorithm and policed with token bucket algorithm to guarantee the bandwidth reservation for each individual flow [19]. ARINC 664 utilizes the virtual link (VL) concept that provides logical communication channels between a source node to one or more destination nodes. ARINC 664 classifies each VL into two priority levels called high priority and low priority. The strict priority scheduler is used to determine transmission order between these priority levels. The limited class number and the strict priority are two main drawbacks that restrict the mixed-criticality for ARINC 664 [20]. Thus, improving the mixed-criticality support for the ARINC 664 by increasing the number of queues and utilizing additional mechanisms is an important research topic [21] [22] [23].

Furthermore, in traditional ARINC 664 networks, the assignment of VL flows to priority classes are done at the offline network planning stage, and there is no concept of priority re-assignment of a VL flow to a different priority class. However, the bandwidth utilization rates of priority classes can vary significantly from time to time within the same flight duration due to the dynamic reconfiguration concepts in the IMA systems [24] [7].

This thesis proposes a priority re-assignment concept that provides significantly lower delay bounds. Note that the decision of flow re-assignment to a new priority class is done by calculating the high and low priority class worst-case delays using the network calculus framework.

1.1 Literature Review

In this section, we present the existing studies on the delay analysis of the network protocols that support mixed-criticality. Firstly, we review the studies about ARINC 664 standard, its advantages, and disadvantages in terms of mixed-criticality. Afterward, the studies about time-triggered protocols such as TTEthernet and IEEE TSN time-aware shaper (TAS) are handled in terms of mixed-criticality. Finally, we summarize the studies about the mixed-criticality performance of the IEEE TSN shaping algorithms and combine these algorithms with the ARINC 664.

ARINC 664 is an event-triggered protocol that is inherently not a suitable standard for mixed-criticality since only two priority levels, namely high priority and low priority, are defined. However, [15] proposed an extended priority scheduler that enhances the schedulability of VLs and provides better mixed-criticality support. They utilize Genetic Algorithm (GA) to optimize the worst-case delay in terms of maximum, average, and standard deviation by considering all VLs used in the network. The numerical results demonstrate that ARINC 664 can satisfy the mixed-critical system requirements by increasing the priority level and assigning the VLs to the defined priority levels in an optimized way.

[25] proposed stand-alone usage of deficit round robin (DRR) and the combined usage of DRR and strict priority (SP) policy instead of using strict priority alone to provide more fairness for the lower priority classes in terms of bandwidth sharing. They assess this proposal with a realistic aviation case study where 104 end systems, 8 switches, 974 VLs, and 6501 delay constraints are used. Their observation demonstrates that the use of stand-alone DRR causes inefficient resource usage but, combined usage of SP and DRR (SP/DRR) provides a more acceptable result in terms of resource utilization. However, it is significant to note that both suggestions increase the latency for

time-critical VL flows compared with strict priority alone. Besides, stand-alone usage of DRR cannot fulfill all latency requirements even with gigabit Ethernet, whereas SP/DRR can satisfy the requirements with gigabit Ethernet restriction. However, even SP/DRR is not a good candidate for mixed-criticality because of this restriction.

[26] proposed a new architecture for the switch output port to allow mixed-criticality support for ARINC 664. The proposed architecture contains a priority level queue for each VL, and these queues are grouped into high priority queues and low priority queues. Bandwidth allocation gap (BAG) based scheduling algorithm is utilized among the priority queues containing time-critical high priority VL flows, while round-robin scheduling algorithm is used for the multiplexing among lower priority queues. Afterward, a non-preemptive strict priority scheduler integrates the frames from high and low priority queues. They evaluate this suggestion with both simulation and network calculus framework. Although the obtained simulation and theoretical delays are within an acceptable range, it is a significant drawback that the memory usage on each output port of the switch increases significantly with the increase in VL numbers.

TTEthernet is one of the time-triggered standards that support the mixed-criticality where safety-critical frames and non-safety-critical are combined in a single network in a deterministic way [27]. In [28], a TTEthernet based simulation environment for the distributed real-time system was introduced. The simulation environment supporting different types of traffic namely time-triggered traffic, rate constrained traffic, and best-effort traffic was used to simulate the automotive network. In the example scenario, rate-constrained traffic was used for the communication between the camera, DVD player, and gateway. The sensor, PID controller, and actuator responsible for cruise control communicate with each other via both time-triggered and rate constrained frames. Additionally, a faulty situation was injected by untimely generating rate constrained traffic. The results demonstrate that if the priority level of the frames in charge of cruise control was set as time-triggered, the PID controller was not affected by the injected faulty frames. In contrast, the jitter value for the cruise control traffic was negatively affected by the injected faulty frames if the priority level was set as a rate-constrained. Thus, the study demonstrates that TTEthernet isolates time-critical traffic from other traffic types, providing a suitable environment for mixed-critical applications.

[29] presented an extension of the OMNeT++ framework that contains nodes compatible with TTEthernet. The introduced extension was used to simulate in-vehicle communication that consists of 70 end systems and 7 switches. The paper demonstrates that the latency of the time-triggered frames remains constant regardless

of the utilization rates of the other traffic types with both simulation and mathematical calculation.

[30] proposed formal timing analysis of rate constrained traffic in TTEthernet to determine the worst-case delay. The paper investigates the effects of the integration policies defined in the TTEthernet standard on the worst-case latency. The proposed analysis is evaluated with the realistic scenario, namely Orion Crew Exploration Vehicle, and the authors specified that the TTEthernet is a suitable solution for the mixed-criticality issue.

However, the performance of the TTEthernet protocol totally relies on global time that is distributed with a defined time synchronization protocol. Therefore, the performance will be significantly degraded if time synchronization is broken due to failures. [31] proposed a diagnostic algorithm to detect the faulty nodes and remove them from the synchronization algorithm. TTEthernet defines many other protocols to handle all situations that may cause a fault. However, although these protocols do not cause great complexity when used on their own, the complexity occurs when all of them interact with each other. Moreover, [32] states that the need for a complex timetable to determine the access of TTEthernet frames to the output port decrease the modularity since adding, deleting, or modifying a flow can affect time table of the whole system. Additionally, since TTEthernet focuses on the best service guarantee for time-triggered frames, the other traffic types (Rate constrained and best-effort) suffer from fairness.

IEEE TSN/TAS is another time-triggered protocol that provides mixed-criticality support with multiple traffic classes, e.g. time-triggered traffic, audio-video-bridging (AVB) traffic, and best-effort [33]. [34] proposed a synthesis approach for gate control list of time-triggered frames by taking the worst-case delays of the other flows into account and evaluated this approach with the test cases. Thus, they demonstrated that IEEE TSN could satisfy both time-critical and AVB traffic requirements. However, although IEEE TSN/TAS simplifies the complexity of the timetable that is used to determine the access time of each frame in TTEthernet by using a gate control list (GCL) that schedules the priority queues at the output port, it still suffers from the configuration complexity of GCL.

[33] presented a formal worst-case timing analysis for IEEE TSN/TAS with taking into account all blocking effects of the shaper. They evaluate their study with two different automotive network topologies. Their test cases contained three types of traffic, in order of priority from highest to lowest: control data traffic, general control traffic, and camera traffic. Their experiments demonstrate that IEEE TSN/TAS

provides very low latency if the time synchronization is accomplished correctly. However, they emphasize that the performance dramatically degraded when the time synchronization is lost. Many other studies such as [35], [36], [37], and [38] investigate the effect of the IEEE TSN/TAS over the AVB traffic.

Similar to the comments for the TTEthernet, [32] states that the need for a complex gate control list that determines the access of time-triggered frames to the output port decrease the modularity since any modification of the GCL can affect time table of the whole system. In addition, it is highlighted that the other weakness of IEEE TSN/TAS is that performance is highly dependent on the correct synchronization. They also state that IEEE TSN/TAS defines guard band and strict priority mechanisms so that the time-critical frames can be totally isolated from the lower priority level traffic. However, these mechanisms may cause over-reservation for the time-critical traffic. Thus, IEEE TSN/TAS also does not provide good fairness.

To sum up, providing predictable and guaranteed bounded delay for time-triggered traffic makes both TTEthernet and IEEE TSN/TAS being a good candidate for a mixed-critical system. However, complex fault-tolerant time synchronization protocol, weak modularity, and fairness are the main disadvantages for both TTEthernet and IEEE TSN/TAS. In addition, requiring complex fault-tolerant time synchronization protocol makes the certification process more challenging for the aviation industry. Therefore, many scheduling and shaping algorithms have been proposed for IEEE TSN, such as peristaltic shaper (PS), urgency-based scheduler (UBS), and burst limiting shaper (BLS). All of these algorithms aim to fulfill quality of service (QoS) requirements for each different traffic class without using error-prone time synchronization protocols.

[39] analyses and compares three of the proposed shaping mechanisms, namely burst limiting shaper, peristaltic shaper, and TAS, by considering automotive network requirements. The paper states that even though TAS provides the best latency and jitter performance for time-critical traffic, it dramatically increases both the worst-case delay for the other lower priority traffic and configuration complexity. They also emphasize that both BLS and peristaltic shaper offer good quality service guarantee together with low complexity.

[33] also investigates a formal worst-case timing analysis of IEEE TSN/PS again with considering the blocking effects of the shaper. The results obtained from the realistic automotive test case setup demonstrate that the peristaltic interval causes an initial delay, and this delay always results in lower PS performance compared to IEEE 802.1Q.

[40] proposed a new asynchronous traffic class called UBS to satisfy the mixed-critical

network requirements in an asynchronous way. UBS allows predictable and low bounded delay even at high utilization rates and low complexity without utilizing the global time. However, the paper assumes that the 802.1Qci-perStream Filtering and Policing standard is used to implement the queue selection process before. This assumption significantly reduces the complexity, but UBS does not reduce the complexity if the queue selection process is also considered.

In [20], deficit-round robin, credit-based shaper, peristaltic shaper, and BLS are studied in detail from the mixed-criticality perspective. They demonstrate the advantage of using the BLS together with the strict priority scheduler in terms of complexity, modularity, fairness, and predictability. Afterward, they integrate the BLS in the ARINC 664 standard to obtain an extended ARINC 664 compatible with mixed-criticality. The traffic class that runs the BLS has two different priority levels, and the priority level at a particular time is determined according to the consumed credit of the BLS. They present the formal worst-case timing analyses using a min-plus algebra based network calculus framework and compare these theoretical results with the simulation. They also apply the proposed architecture to the realistic avionics test case. They conclude that BLS significantly improves the mixed-criticality concept of the ARINC 664 standard. However, although their mechanism decreases the worst-case delay of the low priority class, the high priority class may have a higher worst-case delay than the low priority class.

To sum up, many studies have been carried out on many different network standards to find out an optimal solution for mixed-criticality issue. Although each solution provides an improvement in a particular area, we realize that BLS offers an optimum solution when many criteria such as complexity, modularity, fairness, and predictability are considered together. However, as mentioned above, BLS also needs to be optimized to overcome the contradiction that the worst-case delay of the low priority class may have a lower worst-case delay than the high priority class.

1.2 Objective of the Thesis

Mixed-criticality is an important and timely concept for modern avionics systems that require low bounded latency and jitter for time-critical frames and high bandwidth for the other priority levels. ARINC 664, which is a widely used avionics network protocol, is preferred due to its simplicity. However, it suffers from limited class number and the strict priority scheduler, which limit the mixed-criticality support. Although the time-triggered protocols such as TTEthernet and IEEE TSN provide better mixed-criticality support, they require complex time synchronization mechanisms that

increase certification effort. The objective of this thesis is to improve the schedulability and the mixed-criticality support of ARINC 664 standard without utilizing complex time synchronization mechanisms.

1.3 Hypothesis

In this thesis, we propose a novel dynamic priority assignment (DPA) concept to improve the schedulability and the mixed-criticality support of ARINC 664 standard. The DPA concept is used together with both stand-alone strict priority and the burst limiting shaper (BLS) from the IEEE TSN standard. The DPA is applied on two different case studies called single-hop and multi-hop; the single-hop case study is used to verify the effectiveness of the proposed algorithm, while the multi-hop use case is used to analyze the DPA on a more realistic network topology. In the proposed models, there are three different classes, namely high priority (HP) for safety-critical traffic (SCT), low priority (LP) for rate constrained (RC) traffic, and best-effort (BE) traffic. The class-based forwarding behavior at each node provides the latency and jitter bounds per class, and these bounds can significantly change if the amount of traffic for each class is time-varying. This study utilizes the DPA concept to modify the initial assignment of the virtual link (VL) flows to the priority classes as the class utilization rates change. The priority re-assignment is performed carefully such that the deadline constraints of the HP and LP flows are simultaneously considered. The numerical results show that the class utilization rates can be significantly increased by using the DPA concept with and without the BLS. The simulation is also performed for each use case in order to verify the network calculus result. Both network calculus and simulation results demonstrate that the DPA can improve the schedulability and mixed-criticality of ARINC 664 without using any time synchronization mechanism.

Furthermore, in traditional ARINC 664 networks, the assignment of VL flows to priority classes are done at the offline network planning stage, and there is no concept of priority re-assignment of a VL flow to a different priority class. However, the bandwidth utilization rates of priority classes can vary significantly from time to time within the same flight duration due to the dynamic reconfiguration concepts in the integrated modular avionics (IMA) systems [24] [7]. This study also suggests that the priority re-assignment can potentially prevent undesired delay performances, especially for the lower priority flows.

1.4 Organisation of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we provide the background information that can make the concept in the thesis easier to understand by presenting IEEE TSN, ARINC 664, and network calculus briefly. Chapter 3 provides single-hop and multi-hop system models, network calculus for burst limiting shaper, and the proposed algorithm for both single-hop and multi-hop use cases. In Chapter 4, we present numerical and simulation results for both single-hop and multi-hop network topologies. Finally, we conclude the thesis with Chapter 5.



2 BACKGROUND

In this section, we briefly describe the standards, algorithms, and mathematical backgrounds that we utilize throughout the thesis. Firstly, we present the time sensitive network (IEEE TSN). Then we mention the ARINC 664 standard in detail. Afterward, we conclude this part with the network calculus framework that allows us to define formal timing analysis.

2.1 Time Sensitive Network

In this section, we introduce the IEEE TSN which provides mechanisms to enable high-performance time sensitive transmission over Ethernet-based network. IEEE TSN implemented entirely at the OSI layer 2 (data link layer) provides mechanisms that enable high-performance time sensitive transmission over Ethernet-based network. IEEE TSN also provides real-time deterministic communication with bounded network delay, low delay variation (jitter), and low packet loss [17]. IEEE TSN aims to combine many separate communication standards such as PROFIBUS, PROFINET, ARINC 664, and audio-video bridge (AVB) that are currently used in different sectors

Table 2.1 IEEE TSN standards [17]

IEEE 802.1 STANDARDS	TITLE
802.1 AS-Rev	Timing and Synchronization for Time-Sensitive Applications
802.1 Qbv	Enhancements for Scheduled Traffic
802.1 Qbu	Frame Preemption
802.1 Qca	Path Control and Reservation
802.1 CB	Frame Replication and Elimination for Reliability
802.1 Qcc	Stream Reservation Protocol
802.1 Qci	Per-Stream Filtering and Policing
802.1 Qch	Cyclic Queuing and Forwarding
802.1 Qcr	Asynchronous Traffic Shaping
802.1 CS	Link-local Registration Protocol

[41]. Therefore, IEEE TSN features are targeted at automotive, industrial, AVB and aviation sectors with an open standard. IEEE TSN is composed of a set of standards that are developed and published by the IEEE TSN task group. The list of the standards that address different network problems is represented in Table 2.1. It is important to note that users can decide to implement any combination of the standards based on their requirements of the applications.

802.1AS-Rev standard is responsible for timing and synchronization, which are two crucial mechanisms for the applications requiring deterministic communication. This standard is an OSI level 2 profile of the IEEE 1588 Precision Time Protocol (PTP), which provides synchronization between the IEEE TSN nodes. The standard includes many procedures that are used to ensure the synchronization requirements of the applications, such as time sensitive industrial control systems, autonomous driving systems, and audio-video systems [42].

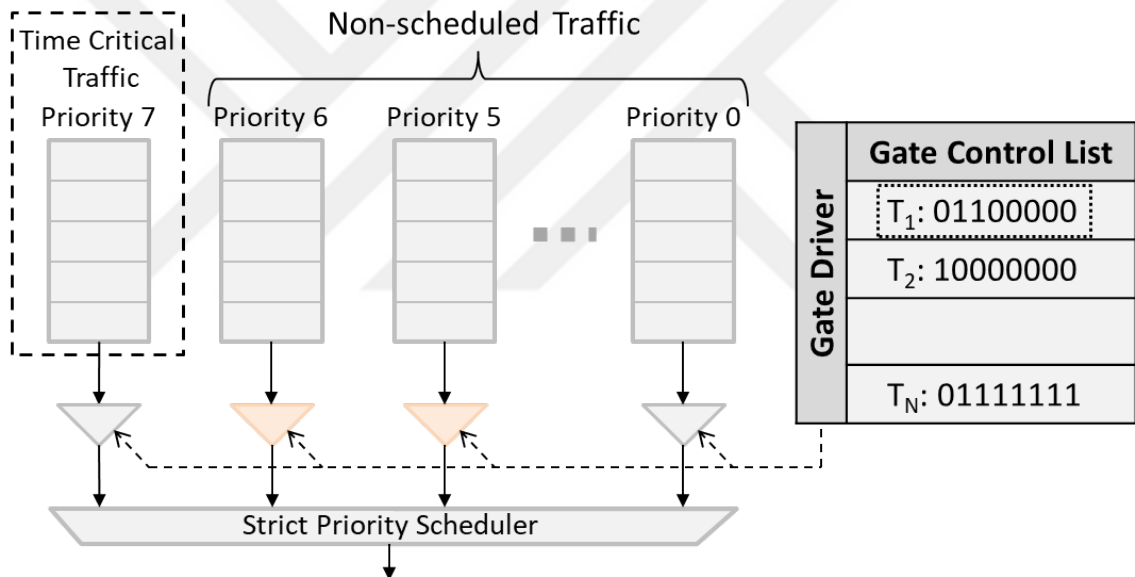


Figure 2.1 Time aware shaper gating mechanism [41]

One of the use cases of 802.1AS-Rev is the other IEEE TSN standard, 802.1Qbv (Enhancements for Scheduled Traffic). This standard includes the time-aware shaper (TAS) that defines time-aware schedule-driven communication. The architecture of the IEEE TSN output node is represented in Fig. 2.1. There are eight different traffic queues whose quality of service (QoS) requirements are different from each other. The principle of the TAS mechanism is that the overall schedule is adjusted by the network planner, i.e., the time slots for each traffic queue are determined for each node being part of the transmission [43]. These time slots are written into the gate control lists that are used to determine whether the state of the time-aware gate is open or closed. Thus, the access of the outgoing queues to the node port is controlled. The

scheduler selects and transmits a frame from one queue whose time-aware gate is open according to the specified scheduling algorithm. In general, the time aware gate of the queue buffering the time-triggered frames is opened alone to avoid congestion and provide deterministic delay for those frames. The gating mechanism allowing isolation between different queues enables the consolidation of different networks into a single network. Thus, the IEEE TSN together with 802.1Qbv is a suitable candidate for the systems requiring improved mixed-criticality support. However, requiring a complex fault-tolerant time synchronization makes the certification process more challenging for 802.1Qbv [44]. In addition to this, the performance will be significantly degraded if time synchronization is broken due to failures [16].

In addition to the TAS, the TSN working group has also proposed two other shaping mechanisms called peristaltic shaper (PS) and Burst limiting shaper (BLS). PS aims to decrease the end-to-end delay of the safety-critical frames by minimizing the elapsed time of the frames inside the switches [33]. It defines two separate time intervals called odd and even phases. Each received safety-critical frame is tagged as either odd or even according to their arrival time slot. The safety-critical frames received at an odd interval must wait till the successive even interval to be transmitted. Similarly, the safety-critical frames received at an even interval cannot be transmitted until the next odd interval starts. The duration of the even and odd phases directly affects the elapsed time for a frame passing through the switch and needs to be determined for each individual switch in a network by utilizing the global time [39]. The interfering traffic may force the safety-critical frames to wait more than one peristaltic phase. In this case, the safety-critical frames on hold are immediately transmitted after the interference ends. The PS provides good predictability for safety-critical frames and acceptable fairness for the lower priority classes. However, the need for global time increases the complexity due to the complicated mechanisms required for time synchronization [20]. Furthermore, it also negatively affects the modularity since adding new flows or deleting flows from the network requires calculating the duration of odd and even phases.

The BLS limits the burstiness of the class it shapes in order to provide mixed-criticality support by offering high fairness for the lower priority level classes [32]. The traffic class that runs the BLS has two different priority levels, and the priority level at a particular time is determined according to the consumed credit of the BLS. The maximum credit threshold, minimum credit threshold, and reserved bandwidth parameters, which are denoted as L_M, L_R , and BW respectively, are used to determine the current priority level of the shaped class. The credit increases with a rate of I_{send} when packets belonging to the shaped class are sent; otherwise, it decreases with the rate of I_{idle} . If the credit reaches the maximum threshold values (L_M),

under the IEEE 802.1 Qci Per-Stream Filtering and Policing standard. These mechanisms improve the quality of service by abstracting the other flows from inappropriate ones [48]. IEEE 802.1 Qch Cyclic Queuing and Forwarding aims to minimize the elapsed time of a frame inside the switch by maintaining at least two output queues for each port that can be enabled and disabled based on a period schedule [49]. IEEE 802.1 Qcr Asynchronous Traffic Shaping provides bounded low latency transmissions without utilizing the time synchronization protocols that increase the complexity dramatically [50]. 802.1 CS Link-local Registration Protocol allows replication of a registration database between the point-to-point connected devices. The size of the replication data is up to 1 MB [17].

In conclusion, IEEE TSN is a standard composed of sub-standards that the TSN working group develops. Each sub-standard is defined to overcome different aspects of the deterministic network problems. The user can implement any combination of these sub-standards according to their requirements.

2.2 ARINC 664

ARINC 664 is a standard that provides deterministic behavior for safety-critical data transmission between the avionics subsystems by defining the electrical and protocol specifications [15]. The standard is a limited and profiled version of the IEEE 802.3 network. The ARINC 664 standard offers additional mechanisms that provide guaranteed quality of service (QoS) to overcome the standard Ethernet in-determinism issue. In contrast to the other common protocols used in avionics systems such

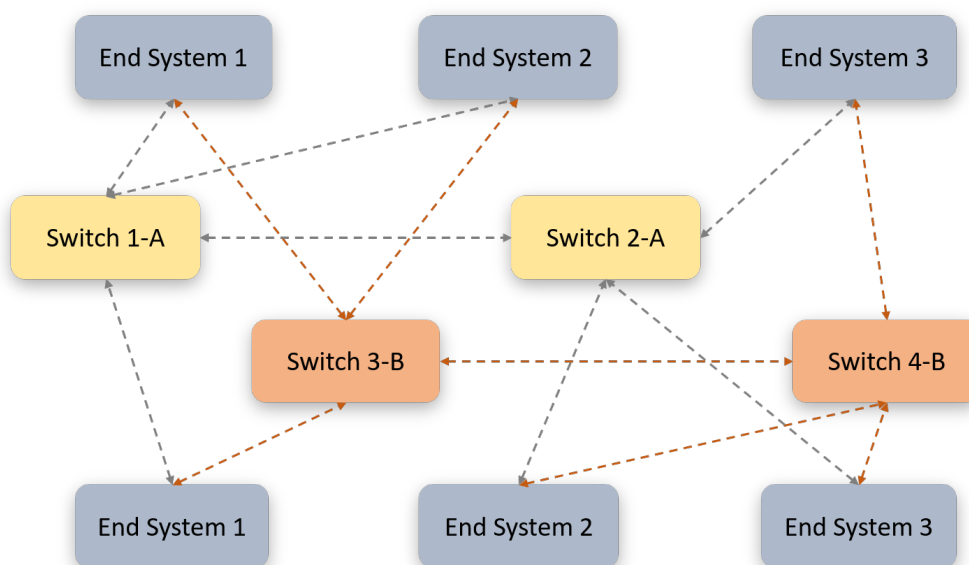


Figure 2.3 ARINC 664 cascade star topology

as ARINC 429 and MIL-STD 1553, Ethernet offers high bandwidth, high flexibility, robustness, and lower cost. Therefore, it has been utilized in many modern avionics platforms so far, such as Airbus A380 and Boeing 787 [51].

ARINC 664 consists of two types of nodes which are end system (ES) and switch. Star topology that supports cascade connection and redundancy is utilized as shown in Fig. 2.3. There are 10/100 Mbps, full-duplex, and point-to-point physical links between the end system and switch nodes. End systems provide a connection between the software and the physical network, whereas the switch basically routes the data to the corresponding destination end system(s) according to the configuration file. The standard relies on the virtual link (VL) concept that provides logical communication channels between a source node to one or more destination nodes [52]. The VL data produced in the application layer or obtained from a sensor is sent to the physical network redundantly through the source end system after being processed by the main mechanisms such as traffic shaping and redundancy management. Then both redundant switches apply VL-based filtering and policing mechanisms before forwarding a frame to the corresponding port(s). The destination end system performs integrity and redundancy mechanisms and transfers the VL data to the upper layers.

In ARINC 664, the flows are shaped based on the leaky-bucket algorithm and policed with the token-bucket algorithm to guarantee the bandwidth reservation for each individual flow. Thus unlike the synchronous protocols, ARINC 664 provides bounded latency in an asynchronous way. Moreover, the redundancy concept provides fault tolerance against network failures and bit errors. Thanks to all of these capabilities, the standard provides determinism, bounded latency, low delay variation (jitter), high reliability, and high availability.

2.2.1 Virtual Link

ARINC 664 standard relies on the Virtual Link (VL) concept. Unlike the ARINC 429 that needs a physical point-to-point connection, the virtual link concept provides separate and isolated uni-directional point-to-point or point to multi-point virtual communication channels over existing physical connections. Thus, the standard allows many virtual connections to share the same physical link.

Each virtual link has two characteristics, namely maximum frame size (MFS) and bandwidth allocation gap (BAG). The BAG represents the minimum time window between two consecutive frames belonging the same VL. These VL parameters are determined by the system integrator during the network planning phase using a network calculus [52].

To sum up, the VL concept allows us to establish a sophisticated and mixed-critical network communication medium without disrupting the determinism.

2.2.2 End System

The end system acts as an interface between the data producer and consumer (application, sensor, or actuator) and the physical network. The end system implements two main functions called transmitting and receiving.

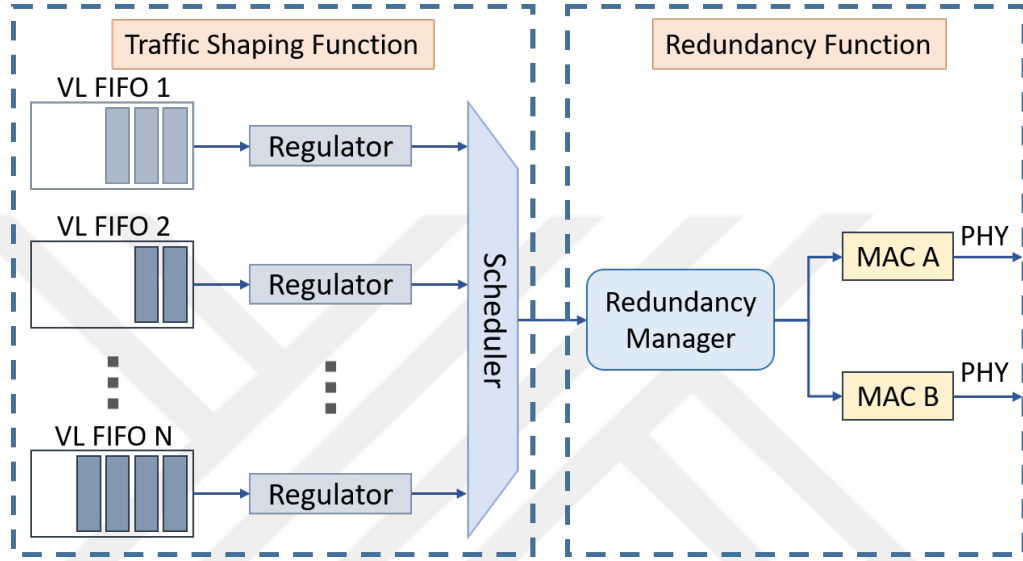


Figure 2.4 Transmitting side of the ARINC 664 end system [53]

In the transmitting part, each VL frame is shaped by VL based traffic regulator and duplicated by the transmitting side of the redundancy manager modules as shown in Fig. 2.4. Each traffic regulator is used to limit the frame rate of each VL in accordance with its bandwidth allocation gap (BAG) value by shaping corresponding VL [53]. Thus, unregulated frames are shaped in such a way that each frame appears right at the beginning of its BAG interval at the output of each regulator when jitter is equal to zero (Fig. 2.5). Then the scheduler selects one of the VL flow coming from

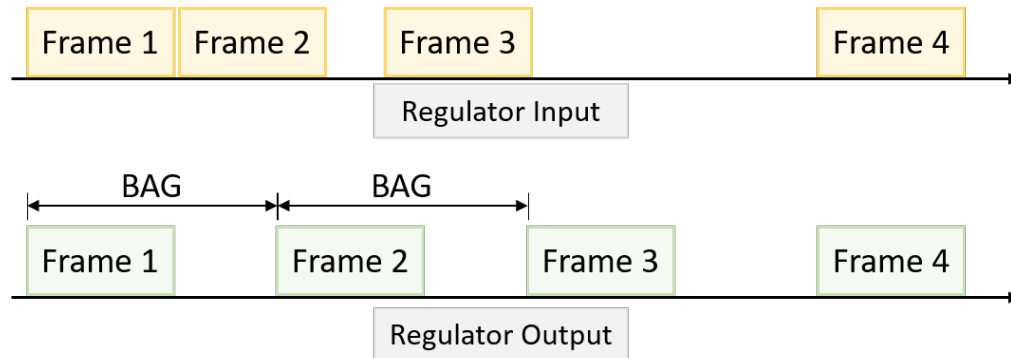


Figure 2.5 Regulating unshaped frames

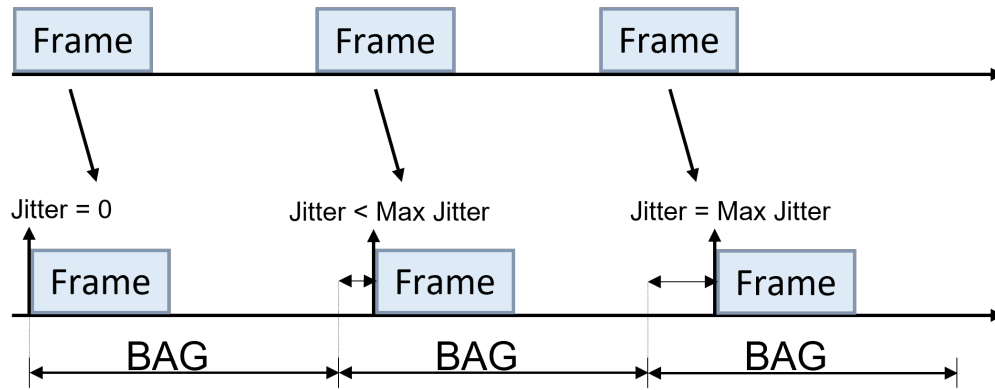


Figure 2.6 Jitter [53]

the regulator and forwards it to the next module called redundancy manager. Note that if more than one frames appear at the input part of the scheduler within the same time window, a certain amount of jitter can be introduced, as shown in Fig. 2.6. The standard prevents the possible traffic bursts by limiting the jitter value with the maximum allowed jitter value. The redundancy manager duplicates each VL and tags each redundant frame with a sequence number per VL. Each copy is forwarded to the corresponding transceiver in order to be transmitted to the switch. Note that one of the other capabilities of the redundancy manager is transmitting non-redundant VLs that are delivered through only one of the redundant networks.

The receiving part of an end system includes an integrity checker, the receiving side of the redundancy manager, and a demultiplexer, as shown in Fig. 2.7 [19]. The integrity checker module controls the sequence number of each redundant VL frame to identify and eliminate invalid frames and forward the valid ones to the redundancy

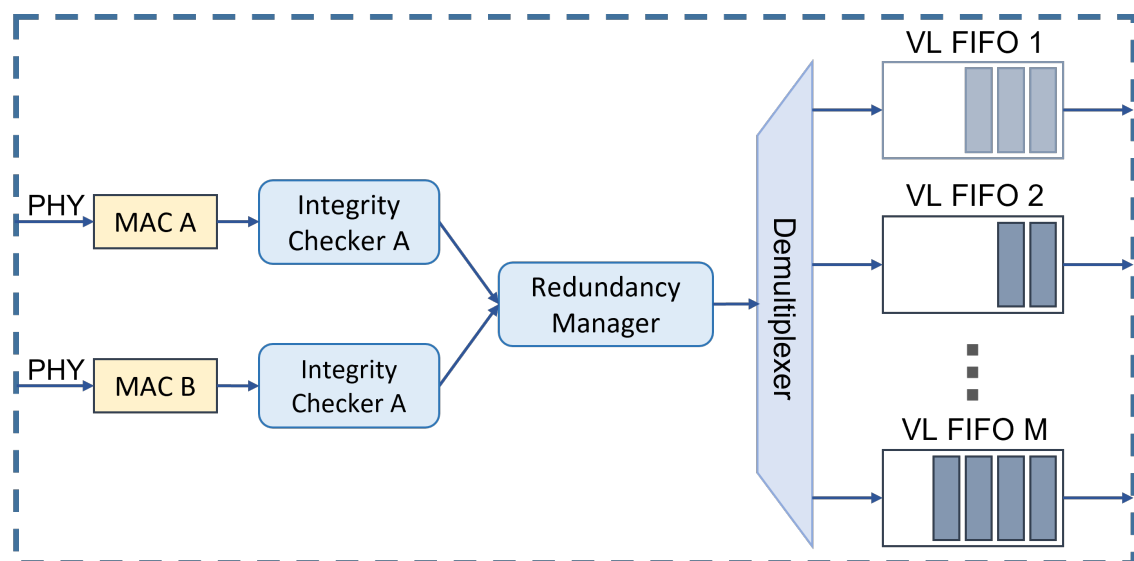


Figure 2.7 Receiving side of the ARINC 664 end system [19]

manager. The redundancy manager module discards the redundant VL frame having the same sequence number incoming within a predefined time window. Demultiplexer transmits the incoming frames into corresponding VL queues.

2.2.3 Switch

An ARINC 664 switch provides a connection between the end systems. An ARINC 664 switch contains five main functions, namely filtering, policing, switching, configuration, and monitoring [12]. Filtering and policing functions inspect the conformity of the incoming frames by comparing frame length and VL-based budget with the values specified in the configuration table [19]. The switch discards the corresponding frame if any violation is detected during filtering operation. The traffic policing function is used to discard VL frames that violate its predefined bandwidth reservation. The standard defines both byte-based and frame-based token bucket algorithm. Thanks to these algorithms, the standard ensures that the maximum burstiness level of each VL is limited. The switching function routes the convenient frames to the corresponding output port(s) in accordance with the configuration table. Configuration function is responsible for capturing the configuration data sent by the network management function and providing them into corresponding tables. The monitoring function logs the statistic data and failure events that occurred during operation. The block diagram of the ARINC 664 switch is shown in Fig. 2.8.

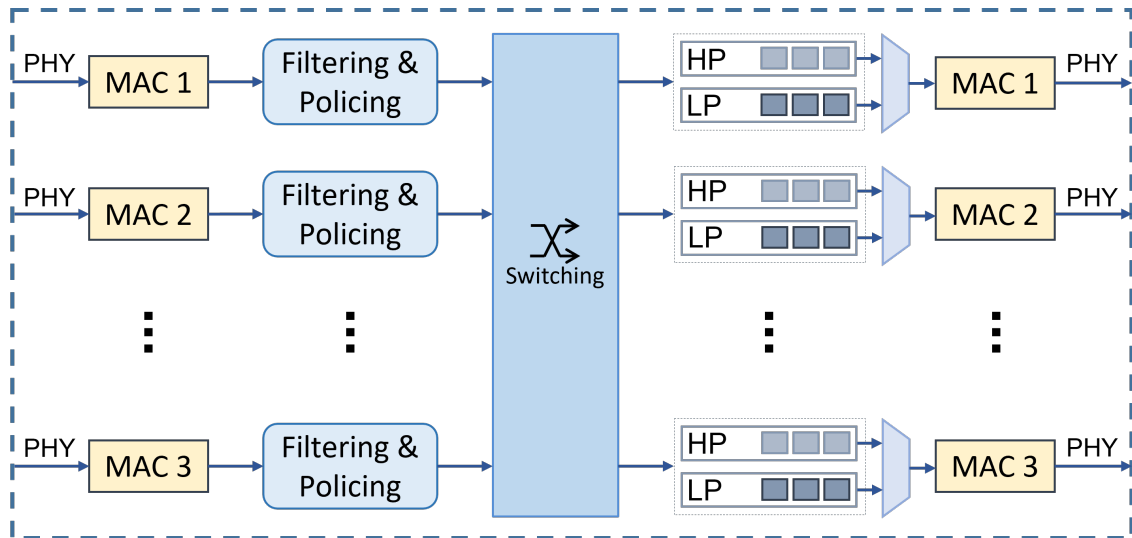


Figure 2.8 ARINC 664 switch block diagram

The ARINC 664 standard defines two priority levels for each VL traffic, namely high priority (HP) and low priority (LP). The priority classifier mechanism assigns the frames into one of the traffic queues based on the priority level specified in the configuration table [54]. The strict priority (SP) scheduler determines the

transmission order of the frames queued in these priority buffers. SP scheduler provides the highest service to high priority traffic by transmitting as many frames as possible in a higher priority frame before selecting a frame from the lower priority level [53]. Thus, the standard ensures that the critical VLs having more strict delay requirements are prioritized compared to other traffic. However, the limited priority level and the strict priority restrict the mixed-criticality concept that is significant for modern avionics systems.

To conclude, ARINC 664 is a relatively simple protocol that provides determinism without using time synchronization protocols; instead, it utilizes the leaky-bucket algorithm, the token-bucket algorithm, and some basic filtering mechanisms to isolate the VLs from adversely affecting each other.

2.3 Network Calculus

A deterministic network needs offline planning so that the end-to-end delay and jitter can be bounded within a value specified by system requirements. In literature, many methods such as trajectory approach, modular performance analysis (MPA), and network calculus have been proposed so far to serve this need [32]. In this study, we use the network calculus method that can be considered as formal timing analysis for the deterministic network since it has already been used to certify the ARINC 664. Network calculus that lies in a theoretical background of Min-plus algebra allows us to calculate the upper bound delay [55]. It is widely used to analyze the worst-case delay bounds for deterministic networks, including ARINC 664 or IEEE TSN [21] [56]. The following subsections provide the network calculus analysis of ARINC 664 for both switch and end system.

2.3.1 Network Calculus Analysis for Switch

In this section, we provide the network calculus analysis of the ARINC 664 switch. First of all, we consider the network calculus analysis for a flow crossing a switch. We assume that there is one-to-one mapping from virtual link n (VL_n) to flow n (f_n) which can be defined by two parameters, namely bandwidth allocation gap (BAG_n) and maximum frame size (s_n). The arrival curve of f_n , which is regulated by the leaky-bucket algorithm, can be formulated as:

$$\alpha_n(t) = r_n t + s_n, \quad (2.1)$$

where $r_n = s_n/BAG_n$ represents the arrival rate. Each output port of an ARINC 664 switch provides a service curve that can be expressed as:

$$\beta_n(t) = R[t - T]^+, \quad (2.2)$$

where R is the service capacity and T is the technological latency. The maximum distance between the arrival curve $\alpha(t)$ and the service curve $\beta(t)$ gives the worst-case delay encountered at the ARINC 664 switch output port. Fig. 2.9 demonstrates a representative example for the worst-case delay. The worst-case delay encountered at the output port of the ARINC 664 switch can be formulated as:

$$D_{SW}^{f_n} = \frac{s_n}{R} + T, \quad (2.3)$$

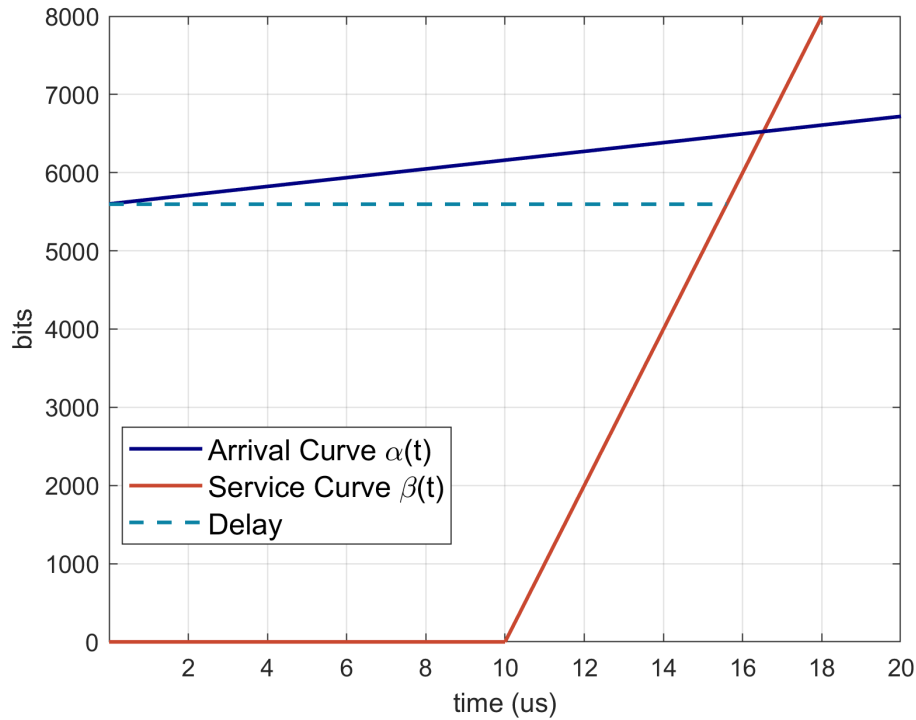


Figure 2.9 Calculating the worst-case delay

The flow arrival curve is modified after crossing a switch. The modified arrival curve α_n^* can be expressed as:

$$\alpha_n^*(t) = \alpha_n(t) \oslash \beta_n(t), \quad (2.4)$$

where $\oslash$ denotes the min-plus deconvolution operator. After applying the deconvolution operation, the equation for modified arrival curve can be expressed as:

$$\alpha_n^*(t) = r_n t + s_n + r_n T, \quad (2.5)$$

Secondly, we provide the network calculus analysis for the scenario that multiple flows share a single switch. Sharing the same output port reveals the concept of residual service curve, which allows us to find the guaranteed individual remaining service capacity for each flow [55]. The residual service curve needs to be calculated based on the service type of the service element. There are two types of residual service curves which are First-In First-Out (FIFO) service type and strict service element. FIFO is used for the flows having the same priority levels, whereas the strict service curve assumption is utilized for the strict service type. Let $p(i)$ represents the priority level of flow i , and $p(i) < p(j)$ denotes that flow i has higher priority than flow j . Under the strict (blind) service type assumption, the residual service curve for f_n is expressed as:

$$\beta_n(t) = R_n[t - T_n]^+, \quad (2.6)$$

where R_n is the remaining service capacity of f_n . R_n can be expressed as:

$$R_n(t) = R - \sum_{\forall i, p(i) < p(j)} r_i, \quad (2.7)$$

where r_i represents the arrival rate of i^{th} flow. The T_n is the latency parameter and is defined as:

$$T_n = \frac{RT + \sum_{\forall i, p(i) < p(j)} s_i + \max_{\forall i, p(i) > p(j)} s_i}{R - \sum_{\forall i, p(i) < p(j)} r_i}, \quad (2.8)$$

where s_i represents the maximum frame size of i^{th} flow.

Considering the priority level and preemption mechanism also affect the worst-case delay analysis. The priority mechanism allows the high priority flows to be served before the lower priority level. Thus, the flows having lower priority level wait till all the high priority flows which are queued in separate FIFO are served. Moreover, the preemption mechanism allows the higher priority flow to interrupt lower priority flow that is currently sending. This mechanism reduces the worst-case delay for high priority flows while the worst-case delay for lower priority flows is negatively affected.

If the preemption mechanism is enabled, the latency parameter T_n is defined as:

$$T_n = \frac{RT + \sum_{\forall i, p(i) < p(j)} s_i}{R - \sum_{\forall i, p(i) < p(j)} r_i}, \quad (2.9)$$

2.3.2 Network Calculus Analysis for End System

In this section, the network calculus analysis of the ARINC 664 end system is provided. The end system utilizes the leaky-bucket algorithm for traffic shaping and strict priority scheduler for multiplexing operation. Thus the mathematical equation for the residual service curve of the ARINC 664 end system is the same with the Eq. (2.6).

In summary, network calculus is a theoretical framework that allows us to analyze network performance. In this study, we utilize network calculus to calculate the worst-case delay analysis.

3

SYSTEM MODEL

This chapter presents the system models for single-hop and multi-hop scenarios, the proposed algorithm, and the network calculus equations that are used to analyze these proposed algorithm and models. First of all, we present the single-hop system model in detail. Afterward, we introduce a novel dynamic priority assignment (DPA) concept that can be used with both stand-alone strict priority and the burst limiting shaper (BLS) from the IEEE TSN standard. Then, we present the network calculus equations that are used for formal delay analysis. Finally, we mention the multi-hop system model.

3.1 Single-Hop Model

This section introduces the proposed single-hop system model depicted in Fig. 3.1. The proposed system model contains N HP flows, M LP flows, and BE traffic sharing the same output port of a node (i.e., ARINC 664 switch or end system). Here, each VL corresponds to an application flow in the ARINC 664 system.

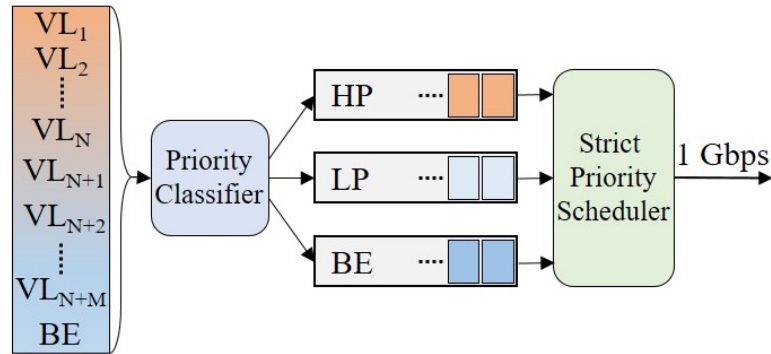


Figure 3.1 Single-hop system model

For simplicity, there are three different priority classes in the proposed system model, including HP, LP, and BE. However, the proposed model can be extended if there are more than three classes. We assume that, within the same priority class, a flow with

lower index number has a higher priority than another flow with higher index number. For example, VL_1 has a higher priority than VL_2 , VL_2 has a higher priority than VL_3 , and so on.

The system model can utilize the BLS together with the strict priority scheduler to shape the HP class. The priority level for the HP class at a particular time is determined by comparing the consumed credit with the maximum credit threshold (L_M) and the minimum credit threshold (L_R). The credit increases with a rate of I_{send} when the packets in the HP class are sent; otherwise, it decreases with the rate of I_{idle} . If the credit reaches the L_M , the priority of the HP class becomes lower than the LP class. The priority returns back to the same level only if the credit is equal to the minimum threshold L_R .

3.2 Multi-Hop Model

In this part, we describe the multi-hop system model, which is an illustrative avionics network architecture of AIRBUS A380 [57]. Fig. 3.2 represents the network topology containing 64 identical end systems and 4 identical switches. A group of 16 end systems is connected to a switch, whereas each switch is connected to two neighbor switches. Note that 1 Gbps Ethernet cable is used for each connection. Like the single-hop model, the multi-hop model includes three different priority levels, namely *HP*, *LP*, and *BE* at each output port of the end system and switch nodes. Note that the multi-hop model can utilize for stand-alone strict priority and the BLS together with the strict priority scheduler.

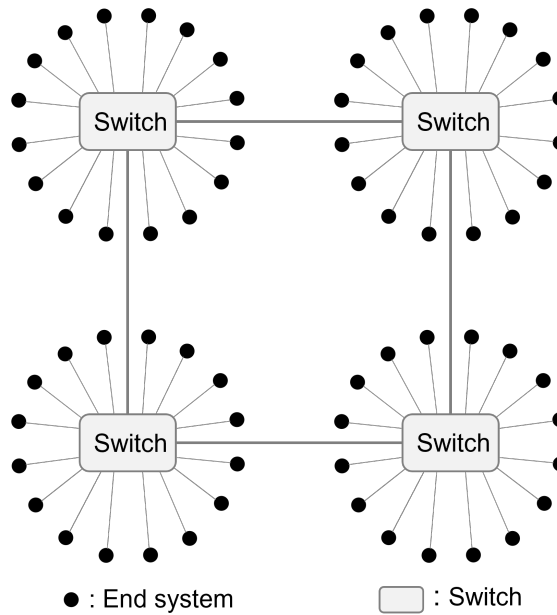


Figure 3.2 Multi-hop system model

The end systems are responsible for both generating and consuming VL flows whose priority levels can be *HP*, *LP*, and *BE*. The VL flows produced by the source end system can be routed to one or more destination end systems through the switch(es) according to the configuration file. The proposed multi-hop system model provides a cascade connection such that a VL flow can be routed over two or more switches. However, we restrict the maximum number of hops with two to avoid the cyclic dependency problem described in [58] [59]. Thus, the proposed model allows VL flow to reach their destination end system(s) after crossing a maximum of two successive switches.

3.3 Dynamic Priority Assignment Algorithm

In this thesis, we propose a dynamic priority assignment concept such that the initial assignment of the flows to priority classes changes according to time-varying class utilization rates. Note that this priority re-assignment is performed carefully such that the highest priority VL flow is moved from the LP to the HP class if the delay constraint of the HP class will not be violated. This is achieved by moving a flow with the lowest index number in the LP class to the HP class since it has the highest priority. Hence, the flow that needs lower delay is assigned to the HP class, assuming that the strict priority scheduler always provides the lowest delay for the HP class. Similarly, the lowest priority VL flow (highest index number) is moved from the HP to the LP class. Hence, the HP class delay can be decreased by sacrificing the delay performance of the lowest priority VL flow. This potentially provides better overall network performance since the sacrificed VL flow may still meet its deadline constraint in the LP class, assuming that the priority levels are determined according to the delay constraints of the VL flows.

Fig. 3.3 shows the flowchart of the proposed algorithm to change the assignment of the VL flows dynamically to the priority classes if needed. Assuming that there are initially N HP and M LP flows, first of all, the maximum worst-case delays of HP and LP classes ($D_{max,HP}$, $D_{max,LP}$) are calculated using the network calculus framework. If $D_{max,HP}$ is higher than $D_{max,LP}$ or maximum delay threshold ($D_{max,Thr}$), the VL flows (starting from the lowest priority, the second lowest priority, and so on) are moved from the HP class to the LP class until the HP class delay constraints are satisfied (i.e., $D_{max,HP}$ is lower than $D_{max,LP}$ and $D_{max,Thr}$). At the second stage, if $D_{max,LP}$ is higher than $D_{max,Thr}$, then the algorithm moves a VL flow from the LP to the HP if the HP class delay constraints are not violated. If one of these constraints is violated, then the algorithm moves this flow back to the LP and terminates. Under the condition that $D_{max,LP}$ is higher than $D_{max,Thr}$, flows can be moved from the LP to the HP until $D_{max,LP}$ is lower than $D_{max,Thr}$ as long as the HP class delay constraints are not violated.

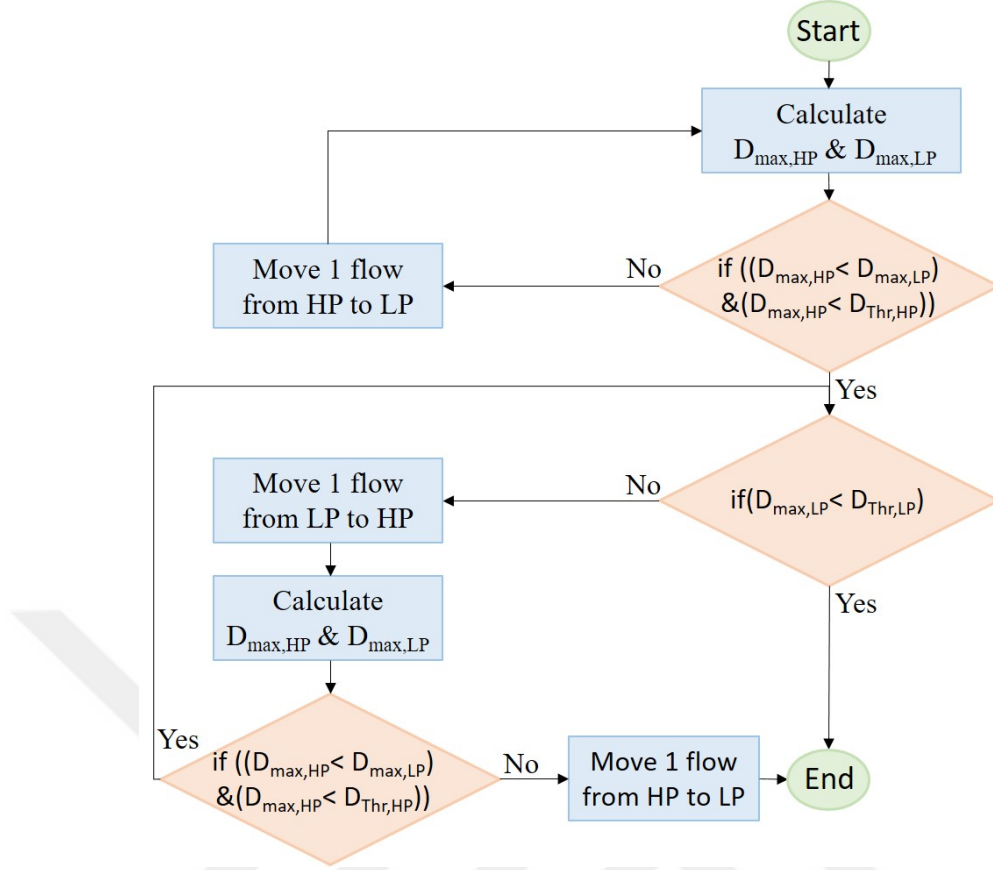


Figure 3.3 Algorithm flow chart

The method without using the algorithm described in the flow chart is called fixed-priority assignment (FPA), where the strict priority scheduler with the HP and LP classes and the leaky-bucket traffic shaper are used as described in the ARINC 664 standard. The method with the algorithm in the flow chart is called dynamic priority assignment (DPA), where the leaky-bucket traffic shaper is used for VLs, but the initial assignment of the VL flows to the priority classes can be changed if there is an assignment that can simultaneously satisfy the objectives of all classes. These two approaches can be applied when the BLS defined in the IEEE TSN standard is used for the HP class in addition to the leaky-bucket shaper for VL flows. Then, these two methods (FPA and DPA) are named BLS-FPA and BLS-DPA, respectively.

Note that the DPA can be directly applied for both single-hop and multi-hop system models. The only thing that changes is the network calculus formulation used for calculating the $D_{max,HP}$ and $D_{max,LP}$. The following section presents how the network calculus can be applied to calculate the worst-case delay bounds for these four methods.

3.4 Network Calculus Delay Analysis

3.4.1 Network Calculus for Single-Hop Model

In this section, we present the worst-case delay analysis for the FPA, DPA, BLS-FPA, and BLS-DPA methods for the single-hop system model.

3.4.1.1 Network Calculus for FPA and DPA

We assume that there is one-to-one mapping from virtual link n (VL_n) to flow n (f_n) which can be defined by two parameters, namely bandwidth allocation gap (BAG_n) and maximum frame size (s_n). The arrival curve of f_n and the service curve of an output port for the ARINC 664 switch can be formulated as defined in Eq.(2.1) and Eq.(2.2), respectively. In this thesis, we utilize First-In First-Out (FIFO) service type for the flows having the same priority level, whereas the strict service type assumption is utilized for the flows having different priority levels. The preemption mechanism is not considered in this thesis. The same notation used in Section 2.3 is utilized, where $p(i)$ represents the priority level of flow i , and $p(i) < p(j)$ denotes that flow i has higher priority than flow j . The residual strict service curve offered by strict priority to flow f_j having the priority level $p(j) \in \{HP, LP, BE\}$ can be expressed as [32]:

$$\beta_j^{sp}(t) = \left[\beta(t) - \sum_{\forall i, p(i) < p(j)} \alpha_i(t) - \max_{\forall i, p(i) > p(j)} s_i \right]^+. \quad (3.1)$$

The worst-case delay bound of f_j can be calculated by obtaining the maximum horizontal distance between the arrival and service curves as[60]:

$$D_j^{sp}(t) = \left(\frac{\sum_{\forall i, p(i)=p(j)} s_i}{R - \sum_{\forall i, p(i) < p(j)} r_i} \right) + \left(\frac{\sum_{\forall i, p(i) < p(j)} s_i + \max_{\forall i, p(i) > p(j)} s_i}{R - \sum_{\forall i, p(i) < p(j)} r_i} \right). \quad (3.2)$$

Note that the above equations can be directly applicable for both FPA and DPA methods since the flow re-assignment to the priority classes affects only the number of flows for HP and LP classes in these equations.

3.4.1.2 Network Calculus for BLS-FPA and BLS-DPA

In ARINC 664, the worst-case delay bounds for LP flows can significantly increase as the bandwidth utilization rate of HP flows increases due to the strict priority scheduler at the switch output port. The BLS concept is applied for ARINC 664 to enhance the

worst-case delay bounds by dynamically changing the priority level of HP class [32]. In other words, the priority level of the HP class can be lower than the LP class for a certain period of time. Window-based approach (WbA) is utilized to determine the priority level of the HP class with respect to the LP class. In this thesis, in addition to applying the BLS-FPA for the HP class, we also utilize the BLS-DPA method presented in Section 3.3.

For the proposed single-hop system model in Fig. 3.1, the minimum service curve for the HP class, where the BLS is utilized, can be expressed as [32]:

$$\beta_{HP}^{bls}(t) = \left(\frac{\Delta_{send}^{min}}{\Delta_{send}^{min} + \Delta_{idle}^{max}} \right) \cdot R \cdot (t - \Delta_{idle}^{max})^+, \quad (3.3)$$

where $\Delta_{send}^{min} = \frac{L_M - L_R}{I_{send}}$ represents the minimum duration of the HP class when it has the highest priority while $\Delta_{idle}^{max} = \frac{L_M - L_R}{I_{idle}} + \frac{s_{LP}}{R}$ represents the maximum duration for the priority level of the HP class when its priority is lower than the LP class. Here, L_M , L_R , I_{send} , and I_{idle} represent the upper credit threshold, the lower credit threshold, the increase rate of the credit, and the decrease rate of the credit for the BLS algorithm, respectively. Note that the BE traffic always has the lowest priority.

Since the service curve of the HP class can take two different values depending on the priority level of the HP class, namely the non-preemptive strict priority service curve and the BLS service curve, the multiplexed service curve can be calculated as:

$$\beta_{HP}^{mux}(t) = \max \left(\beta_{HP,2}^{sp}, \beta_{HP}^{bls} \otimes \beta_{HP,0}^{sp} \right)(t), \quad (3.4)$$

where $\beta_{HP,2}^{sp}(t)$ and $\beta_{HP,0}^{sp}(t)$ represent the service curves of the HP class when it has lower and higher priority than the LP class, respectively, while β_{HP}^{bls} is the service curve of the BLS. Note that $\otimes$ denotes the min-plus convolution operator and β_{HP}^{bls} is given in Eq. (3.3). $\beta_{HP,2}^{sp}(t)$ and $\beta_{HP,0}^{sp}(t)$ can be expressed as follows:

$$\beta_{HP,2}^{sp}(t) = (R \cdot t - \alpha_{LP}(t) - \max_{i \in BE} s_i)^+, \quad (3.5)$$

$$\beta_{HP,0}^{sp}(t) = (R \cdot t - \max_{i \in (LP, BE)} s_i)^+. \quad (3.6)$$

The service curve $\beta_{HP,2}^{sp}(t)$ can be expressed as $R_{HP,2}^{sp} (t - T_{HP}^{sp})^+$, where $R_{HP,2}^{sp}$ represents the residual service capacity while T_{HP}^{sp} includes the technological latency for the case that the HP class has lower priority than the LP class. Similarly, $\beta_{conv}(t)$ can be expressed as $R_{conv} (t - T_{conv})^+$ where R_{conv} denote the residual service capacity while

T_{conv} includes the technological latency for the service curve of $\beta_{HP}^{bls} \otimes \beta_{HP,0}^{sp}$. Therefore, the worst-case delay bounds for the HP class can be expressed as:

$$D_{HP}^{mux} = \min(D_{HP,2}^{sp}, D_{conv}). \quad (3.7)$$

Substituting $D_{conv} = \sum_{i \in HP} \frac{s_i}{R_{conv}} + T_{conv}$ and $D_{HP,2}^{sp} = \sum_{i \in HP} \frac{s_i}{R_{HP,2}^{sp}} + T_{HP,2}^{sp}$ into Eq. (3.7), the worst-case delay bound for the HP class can be calculated.

The service curve of the LP class is also affected when the BLS is applied to the HP class. The multiplexed minimum service curve offered to the LP class can be expressed as:

$$\beta_{LP}^{mux}(t) = \max\left(\beta_{LP}^{sp}, \beta_{LP}^{bls}\right)(t), \quad (3.8)$$

where $\beta_{LP}^{sp}(t)$ and $\beta_{LP}^{bls}(t)$ represent the minimum service curve offered to the LP class when it has lower priority than the HP class and when the BLS is applied to the HP class, respectively. These service curves can be expressed as:

$$\beta_{LP}^{sp}(t) = (R \cdot t - \alpha_{HP}(t) \oslash \beta_{HP}^{bls}(t) - \max_{i \in (HP, LP, BE)} s_i)^+, \quad (3.9)$$

$$\beta_{LP}^{bls}(t) = (R \cdot t - \Upsilon_{HP}^{bls} - \max_{i \in (HP, LP, BE)} s_i)^+, \quad (3.10)$$

where $\oslash$ denotes the min-plus deconvolution operator and Υ_{HP}^{bls} represents the maximum service curve offered by the BLS. It becomes $\Upsilon_{HP}^{bls} = R \cdot t$ when there is no backlogged LP traffic, otherwise:

$$\Upsilon_{HP}^{bls} = \frac{R}{\Delta_{\Upsilon_{SCT}}} \cdot \Delta_{send}^{max} \cdot t + \frac{R}{\Delta_{\Upsilon_{SCT}}} \cdot \Delta_{send,0}^{max} \cdot \Delta_{idle}^{min}, \quad (3.11)$$

where $\Delta_{send}^{max} = \frac{L_M - L_R}{I_{send}} + \frac{s_{HP}}{R} + \min(\frac{s_{LP}}{R} \cdot \frac{I_{idle}}{I_{send}}, \frac{L_R}{I_{send}})$ represents the maximum duration of the HP class having the highest priority. The initial value for Δ_{send}^{max} is represented as $\Delta_{send,0}^{max} = \frac{L_M}{I_{send}} + \frac{s_{HP}}{R}$. In addition, $\Delta_{idle}^{min} = \frac{L_M - L_R}{I_{idle}}$ represents the minimum duration for the HP class when its priority level is lower than the LP class. Here, $\Delta_{\Upsilon_{SCT}} = \Delta_{send}^{max} + \Delta_{idle}^{min}$ denotes the period of the maximum service curve analysis.

The service curve $\beta_{LP}^{\mu}(t)$ can be expressed as $R_{LP}^{\mu}(t - T_{LP}^{\mu})$, where $\mu \in \{sp, bls\}$. R_{LP}^{μ} represents the residual service capacity, while T_{LP}^{μ} includes the technological latency for the case that the LP class has lower priority than the HP class. Therefore, the worst-case delay can be expressed as:

$$D_{LP}^{mux} = \min(D_{LP}^{sp}, D_{LP}^{bls}). \quad (3.12)$$

Substituting $D_{LP}^\mu = \sum_{i \in LP} \frac{s_i}{R_{LP}^\mu} + T_{LP}^\mu$ into Eq. (3.12), the worst-case delay bound for the LP class can be calculated.

The modified arrival curve after crossing BLS is equal to the arrival curve of the strict priority scheduler. This equation can be expressed as:

$$\alpha_{HP}^{sp}(t) = \alpha_{HP}^{*,bls}(t) = \min(\gamma_{HP}^{bls}(t), \alpha_{HP}(t) \oslash \beta_{HP}^{bls}(t)). \quad (3.13)$$

The multiplexed modified arrival curve for HP class can be calculated by combining Eq.(3.13) and Eq.(2.4). Furthermore, since the LP class is not modified with BLS, the arrival curve of the strict priority is exactly equal to the arrival curve of the system model.

Note that the above equations can be directly applicable for both BLS-FPA and BLS-DPA methods since the flow re-assignment to the priority classes affects only the number of flows for HP and LP classes in these equations.

3.4.2 Network Calculus for Multi-Hop Model

This section introduces the worst-case delay analysis of the BLS-FPA and BLS-DPA methods for the multi-hop model. Algorithm 1 represents the pseudo-code that needs to be followed to calculate the worst-case delay of the n^{th} flow (f_n) crossing multiple switch nodes. In this algorithm, first of all, all switches that each VL flows pass through is determined. Then, for each crossed switch, it is checked whether the flows sharing the same output port as f_n have previously passed through a switch. For the flows that have already passed a switch, the modified arrival curve reflecting the effect of the

Algorithm 1 Calculating the worst-case end-to-end delay

```

1: for each crossed switch node  $i$  do
2:   for each flow sharing the same output port with the  $n^{th}$  flow  $f_n$  do
3:     if flow crossed any switch before then
4:       calculate modified arrival curve of the flow;
5:     else
6:       calculate arrival curve of the flow;
7:     end if
8:   end for
9:   calculate residual service curve of the given flow
10:  calculate worst-case delay of the given flow
11: end for
12: calculate the delay due to end system
13: calculate the worst-case end-to-end delay

```

previous crossing is calculated, while for the other flows, the arrival curve is calculated. After calculating all the arrival and modified arrival curves for each flow sharing the same output port, the residual service curve of the flow f_n is calculated. Afterward, the delay for each flow caused by the crossing switches is calculated individually. Then, the delay caused by the end system is calculated. Finally, the worst-case end-to-end delay for each flow is calculated by summing the delays due to all switches that the flow passes through and the end system delay. Note that all formulations used for the multi-hop system model are the same as those used for the single-hop system model. As a result, the worst-case end-to-end delay for BLS-FPA and BLS-DPA methods can be calculated by following the steps in Algorithm 1.



This chapter presents the performance evaluation of the proposed methods on the proposed system models described in Chapter 3 by providing both theoretical and simulation results. First, we present the performance evaluation of FPA, DPA, BLS-FPA, and BLS-DPA for the single-hop system model. Afterward, we present numerical and simulation results for the multi-hop system model.

4.1 Performance Evaluation for Single-Hop Model

4.1.1 Numerical Experiments for Single-Hop Model

In this section, we provide numerical experiments to demonstrate the delay performance of four methods described in Section 3.3 namely FPA, DPA, BLS-FPA, and BLS-DPA. Note that our proposed dynamic priority assignment based methods are DPA and BLS-DPA, while the fixed priority assignment based methods FPA and BLS-FPA are used as the benchmark.

For all experiments, one node with three priority classes including HP, LP, and BE is considered, as shown in Fig. 3.1. The same experiment parameters in [21] are used in this study. The maximum frame size (MFS) and the bandwidth allocation gap (BAG) are set to 64 bytes and 2 ms for SCT and RC flows, respectively, while MFS for BE is 1024 bytes and BAG is 8 ms. In the first set of the experiments, the utilization rate of the HP class (UR_{HP}) is increased from 0.1% to 78% while the utilization rate of the LP class (UR_{LP}) is kept constant as 20%. In the second set of the experiments, UR_{LP} is varied from 0.5% to 72% while UR_{HP} is kept constant as 20%. Note that the utilization rate is increased by adding new flows to the corresponding class and the BE is used to load the system up to the 100% utilization rate. The delay analysis of the BE is not presented since there is no deadline constraint. Without loss of generality, the delay requirements for both HP and LP traffic classes are set to 2 ms. The L_M , L_R , and BW parameters required for the BLS-FPA and BLS-DPA experiments are set to 22118, 0 and 0.46, respectively, as in [21]. The BW parameter of 0.46 means that the

bandwidth of the HP class is set to the 46% of the total outgoing capacity. We report the worst-case (WC) and average (Avg.) delays for all four methods. Since there is no priority re-assignment for FPA and BLS-FPA, the WC and Avg. delay results will be the same. Therefore, only the WC delays are shown for FPA and BLS-FPA in the figures. To quantify the improvement due to the priority re-assignment, the average of the worst-case delays for the flows, that are initially at the same class, is calculated after the priority re-assignment.

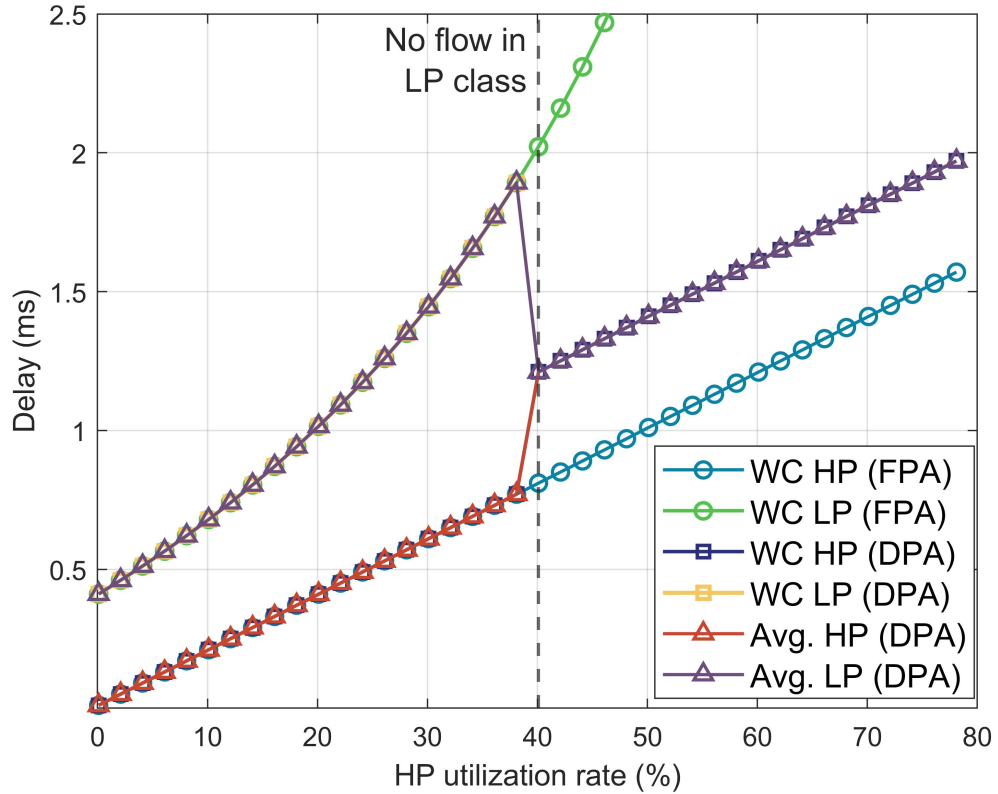


Figure 4.1 The delay performance of FPA and DPA as UR_{HP} varies

Fig. 4.1 illustrates the effect of the HP utilization rate on the delay results for the FPA and DPA methods. The same delay results are observed for both methods until UR_{HP} reaches 38.1%. Beyond this point, the delay requirements of the LP class cannot be satisfied using the FPA while the worst-case and average delays of the LP class significantly decrease by using the DPA method thanks to the priority re-assignment of the initially LP flows to the HP class. Note that all LP flows have to be moved to the HP class for this particular experiment, otherwise some of the LP flows cannot satisfy the LP delay constraint since all LP flows have the same delay constraint of 2 ms. Therefore, the worst-case and average delays for both HP and LP flows are the same when UR_{HP} is beyond 38.1%. As a result, the DPA method can satisfy the delay constraints of HP and LP flows for all UR_{HP} values while the FPA method cannot satisfy the delay constraints of LP flows when UR_{HP} is beyond 38.1%. This improvement for

the LP flows is achieved by increasing the delay results of the HP class in an acceptable manner since the delay requirements of the HP class are still met.

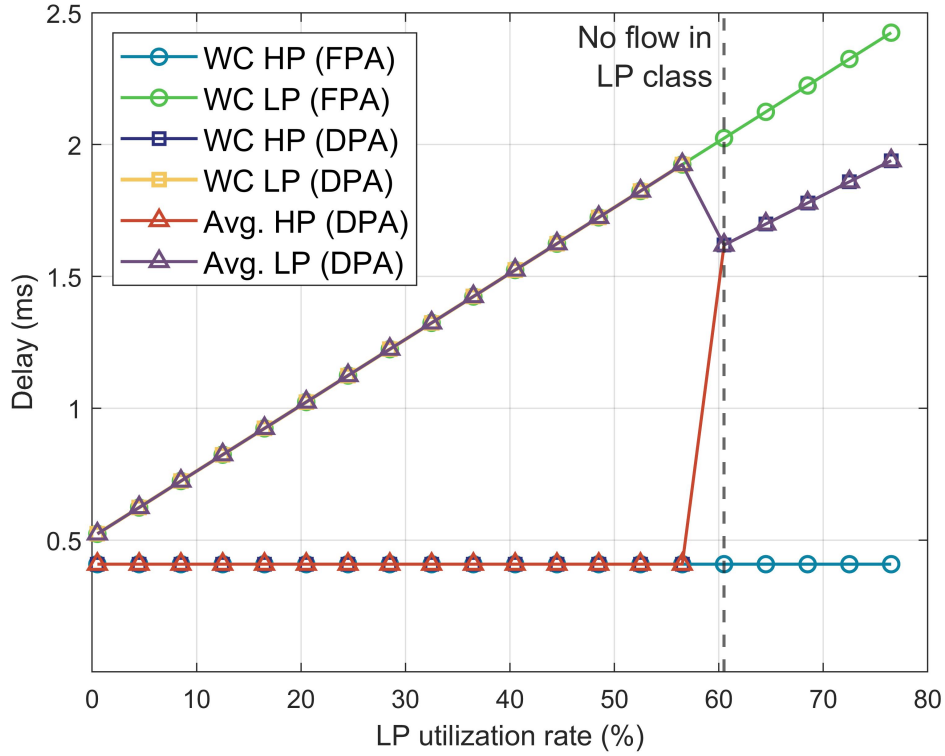


Figure 4.2 The delay performance of FPA and DPA as UR_{LP} varies

Similarly, Fig. 4.2 demonstrates how the change in UR_{LP} affects the worst-case and average delay results of the FPA and DPA methods. Both the FPA and DPA methods satisfy the delay requirements for HP and LP flows when UR_{LP} is lower than 56.5%. Beyond this point, the DPA method performs the priority re-assignment of initially LP flows to the HP class to meet the delay requirement for the LP class. Although this priority re-assignment increases the delay of the HP class, it does not pose a problem since the delay requirements are met for both HP and LP flows. Thanks to the DPA method, a significant improvement in terms of supporting higher LP utilization rates is achieved by fulfilling the delay requirements that cannot be satisfied by the FPA method.

Fig. 4.3 shows the impact of UR_{HP} on the delay results when the BLS-FPA and BLS-DPA methods are utilized. Until UR_{HP} reaches 18.1%, the delay results of the HP class are the same for both BLS-FPA and BLS-DPA and the delay results of the LP class are the same for both methods as well since there is no need for dynamic priority assignment. However, after this critical point, BLS-FPA exhibits an undesirable behavior such that the worst-case delay is higher for the HP compared to the LP. BLS-DPA avoids this problem by priority re-assignment in such a way that some of HP flows (starting from

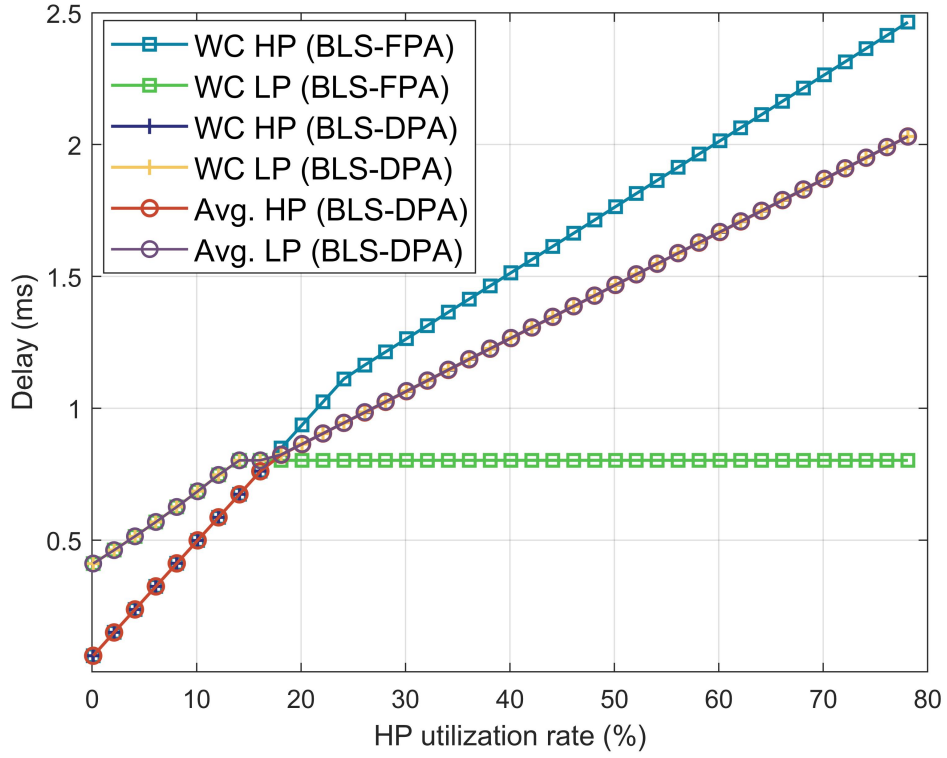


Figure 4.3 The delay performance of BLS-FPA and BLS-DPA as UR_{HP} varies

the lowest priority one) are moved to the LP class until the worst-case delay for the HP class is lower than the LP class. Therefore, the worst-case delays of both classes can be seen at midpoint in Fig. 4.3. The results show that the average delays of HP and LP are also close to each other when UR_{HP} is beyond 18.1%. BLS-DPA outperforms BLS-FPA in terms of schedulability since the maximum UR_{HP} values, for which the deadline requirements of both HP and LP classes are met, are 60.1% and 78.1% for BLS-FPA and BLS-DPA, respectively.

The impact of UR_{LP} on the delay results for both BLS-FPA and BLS-DPA is demonstrated in Fig. 4.4. For low UR_{LP} values, BLS-FPA suffers from the above mentioned undesired problem that the worst-case delay for the HP class is higher than the LP class. However, BLS-DPA resolves this issue by re-assigning the HP flows into the LP class. Both methods provide the same results when UR_{LP} is between 24.5% and 56.5%. After this critical point, the worst-case delay for the LP class exceeds the deadline constraint for BLS-FPA. In contrast, BLS-DPA moves some of the HP flows into the LP class until the delay constraint is satisfied. It is crucial to note that the average delay for the LP class is below the worst-case delay, which means that the priority level is changed only for a few flows. Besides, BLS-FPA schedulability expires when UR_{LP} is equal to 56.5%. On the other hand, there is no restriction for BLS-DPA in terms of schedulability concern.

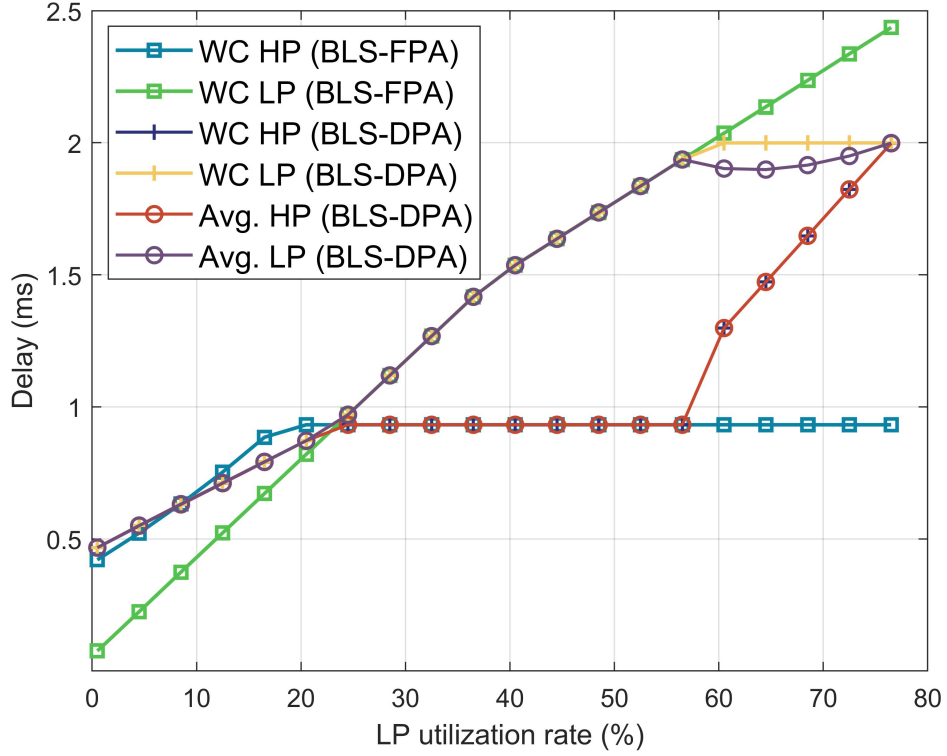


Figure 4.4 The delay performance of BLS-FPA and BLS-DPA as UR_{LP} varies

Finally, it is worth mentioning that DPA yields lower delay results for the HP class compared to BLS-DPA until UR_{LP} reaches 60.1%. However, as UR_{LP} increases beyond this point, BLS-DPA provides the delay results that are more convenient for the HP class. Similarly, as UR_{HP} increases, DPA outperforms BLS-DPA when UR_{HP} is lower than 40.1% while the difference is only a few microseconds for higher UR_{HP} values.

4.1.2 Simulation Analysis for Single-Hop Model

In this section, we present the simulation analysis for the single-hop model described in Section 3.1. The system model is implemented on OMNeT++ (Objective Modular Network Test-bed in C++) simulator to validate the effectiveness of the proposed algorithm. The simulation environment described in [61] is modified to implement the proposed system models that utilize the BLS algorithm.

The same case study described in Section 4.1.1 is utilized. The simulation environment contains 12 source end systems that produce traffic, one destination end system that consumes the traffic, and a switch that provides the connection between the end systems as shown in Fig. 4.5. The source end systems are grouped into three, with each end system group generating only one of the SCT, RC, and BE traffic types. Table 4.1 shows the number of flow generated by each end system group. The flow generation

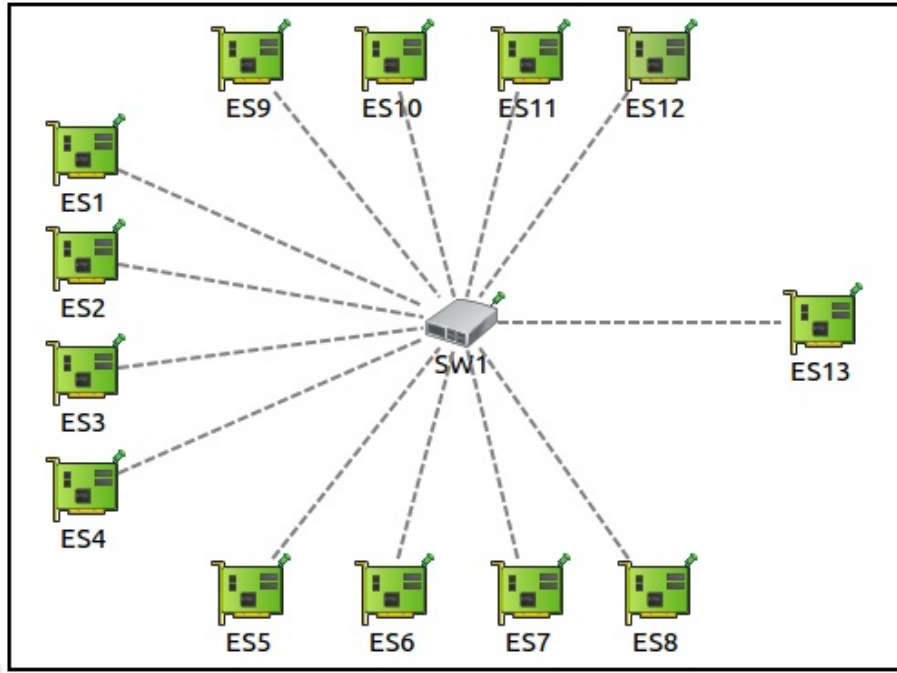


Figure 4.5 Simulation environment for single-hop system model

interval for each end system is randomly determined using normal distribution model with a mean of 2 ms and standard deviation of 0.02 ms. Each simulation scenario is run for 15 seconds and the highest latency values of the classes are considered.

Table 4.1 Number of flows generated by each end system group

	Traffic Type	Scenario 1	Scenario 2
ES1-ES4	SCT	[4:160:3044]	780
ES5-ES8	RC	780	[20:200:3020]
ES9-ES12	BE	load network up to 100%	

The end system model can be examined in two parts which are called as the transmitting part and receiving part, whose components are shown in Fig. 4.6 [61]. The transmitting part contains traffic source that is responsible for creating frames, regulator logic that regulates the bursty traffic by implementing the leaky-bucket algorithm, queues that store regulated traffic in First-In First-Out (FIFO) fashion, the VL scheduler that decides the frames to be transmitted, and the redundancy controller which is in charge-of maintaining the sequence number on per VL basis. The MAC module provides the required infrastructure to transmit frames through the physical layer in a full-duplex way. The receiving part contains the integrity checker module that controls the ordering of the frames on a per-VL basis, the redundancy checker module that allows transmission of only one of the redundant frame, the classifier that classifies the incoming frames into one of the three traffic type, and the traffic sink module that consumes the receiving frames to calculate the performance results

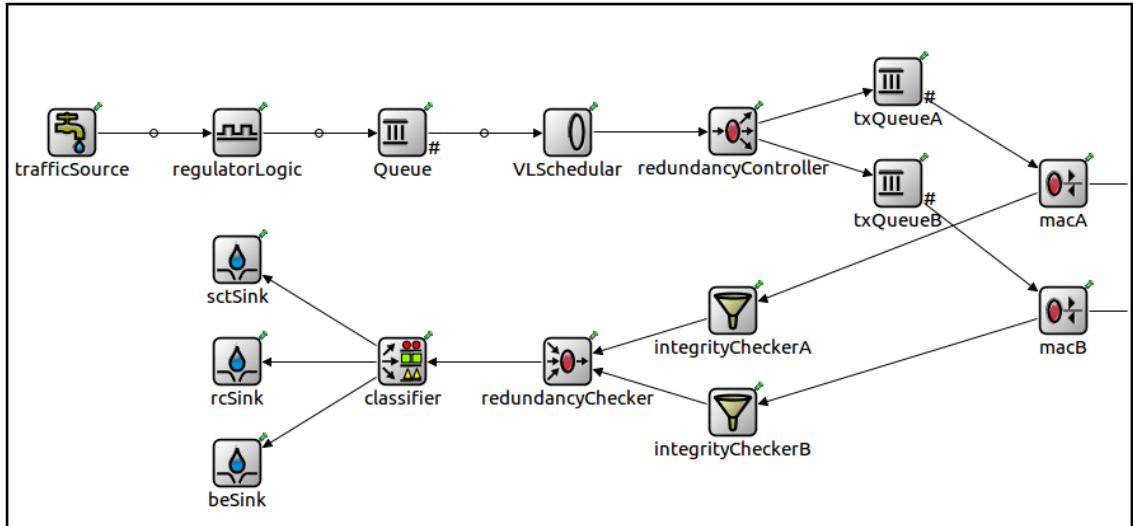


Figure 4.6 End system simulation model inner structure [61]

such as minimum, maximum, and mean end-to-end delay. In the single-hop model, we only focus on the delay caused by the switch itself. Therefore, all the latency parameters used in the end system model are set to zero and the transmission delay of the frames is minimized by increasing the data rate inside the end system modules.

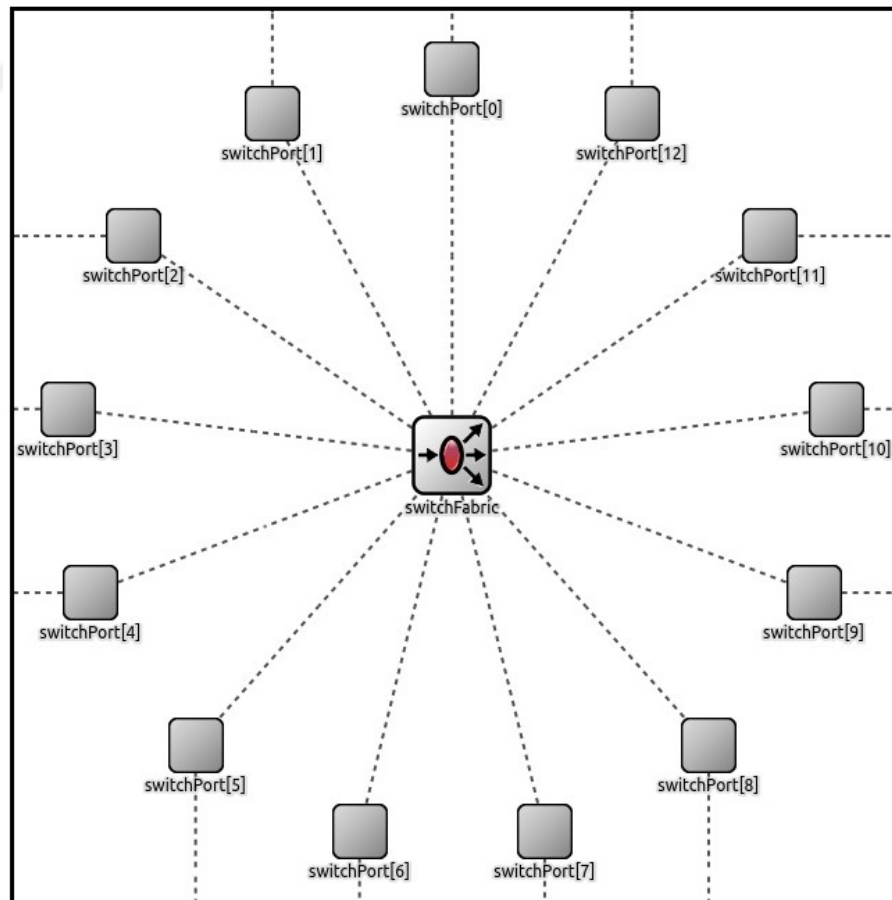


Figure 4.7 Switch module

The switch model contains 13 ports and a switch fabric as shown in Fig. 4.7. The switch port contains priority classifier, three FIFO queues that hold high, low, and best-effort frames, MAC, and frame filtering module as shown in Fig. 4.8. The priority classifier module classifies the incoming frame into three priority queues according to the configuration table. The MAC module contains the BLS algorithm whose algorithm is described in Section 2.1. The priority status of the HP and LP queues is determined according to the current situation of the BLS algorithm. Note that BE always has the lowest priority since it does not have any delay constraint. The frame filtering module discards the invalid frames after checking the size and integrity. It is important to note that the policing module is not implemented to avoid frame drops due to the predefined bandwidth reservation violation.

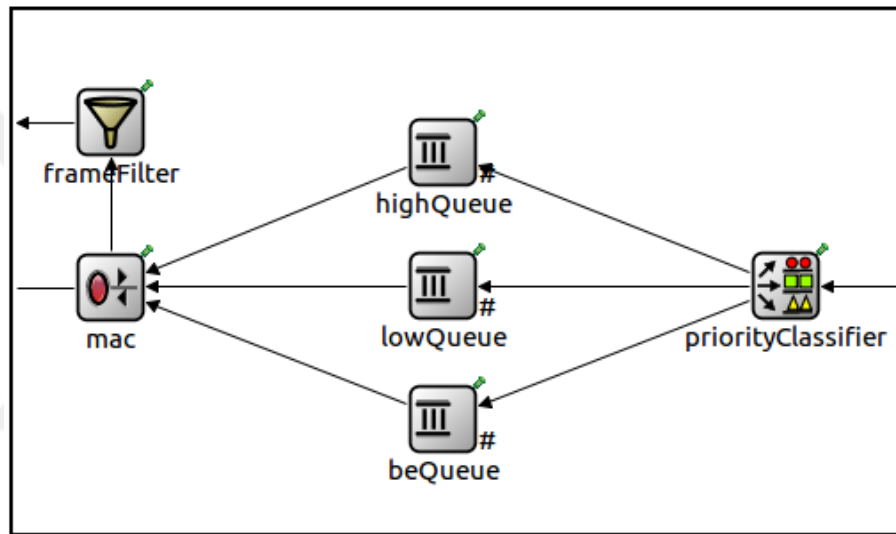


Figure 4.8 The switch port module

The switch fabric module receives the incoming frames and stores them into the Rx Queues as demonstrated in Fig. 4.9. The scheduler utilizes the round-robin algorithm to schedule the Rx queues. The scheduled frames are directed by the router module to the one or more egress ports of the switch according to the VL destination defined

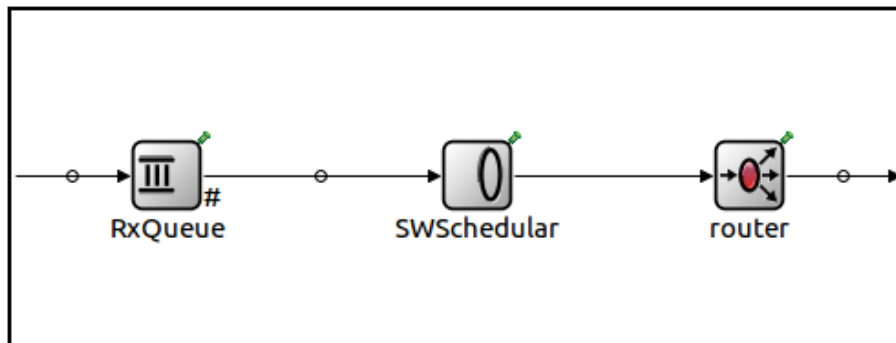


Figure 4.9 Switch fabric module

in the configuration XML (Extensible Markup Language) file.

Fig. 4.10 and Fig. 4.11 demonstrate the comparison of the worst-case and simulation delay results of HP class and LP class, respectively, when the HP utilization rate changes. The delay values obtained from simulation are always lower than the worst-case delay for both BLS-FPA and BLS-DPA methods regardless of the HP utilization rate. Additionally, the simulation delay values and theoretical delay values exhibits similar characteristics. These two outputs demonstrate that the simulation results confirm the theoretical results.

The comparison of the worst-case and simulation delay results of both HP and LP class when the LP utilization rate changes is demonstrated in Fig. 4.12 and Fig. 4.13, respectively. For both BLS-FPA and BLS-DPA methods, the worst-case delay values are always higher than the simulation delay values regardless of the LP utilization rate. As a result, these two graphs confirm the theoretical formulas since the analytical model always gives the upper bound and the characteristic of the simulation and worst-case delay values are similar.

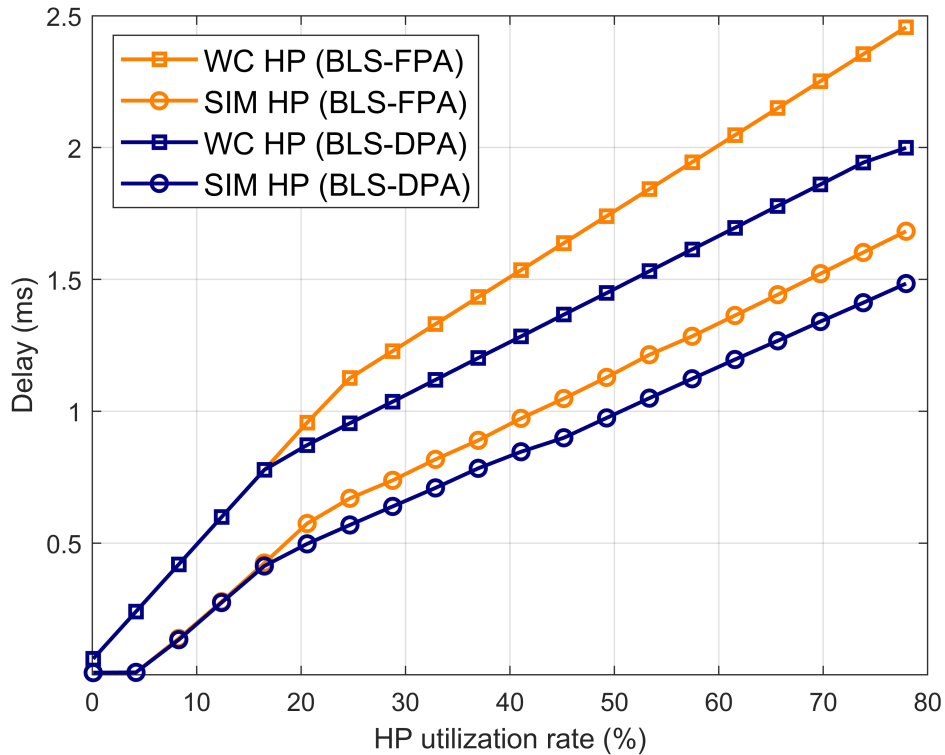


Figure 4.10 The comparison of the worst-case and simulation delay of HP class as UR_{HP} varies (single-hop scenario)

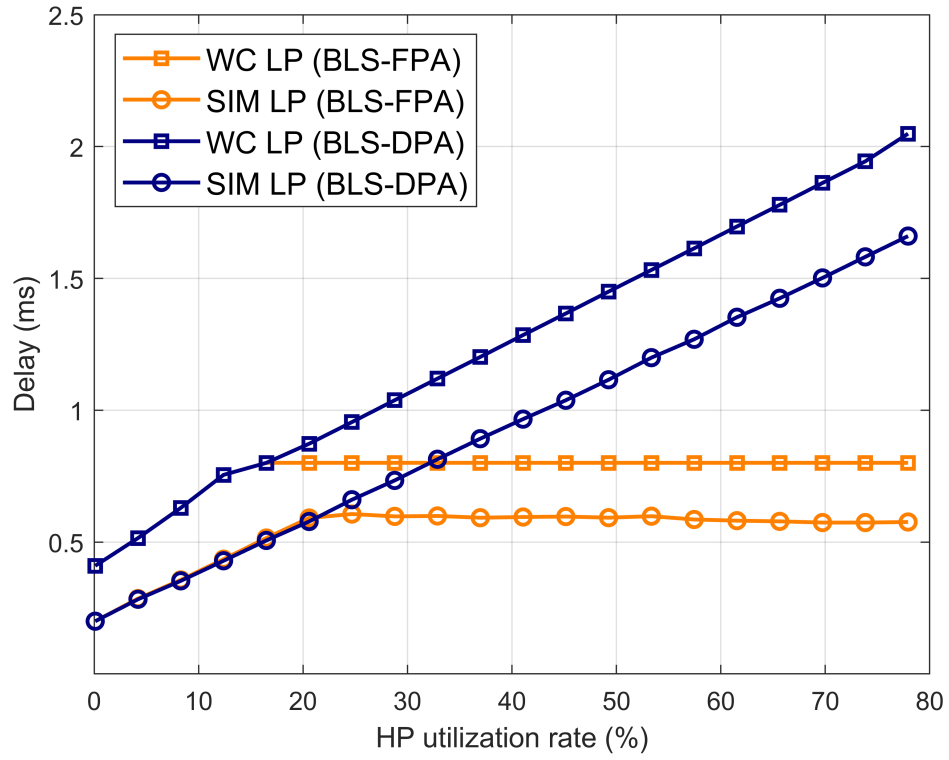


Figure 4.11 The comparison of the worst-case and simulation delay of LP class as UR_{HP} varies (single-hop scenario)

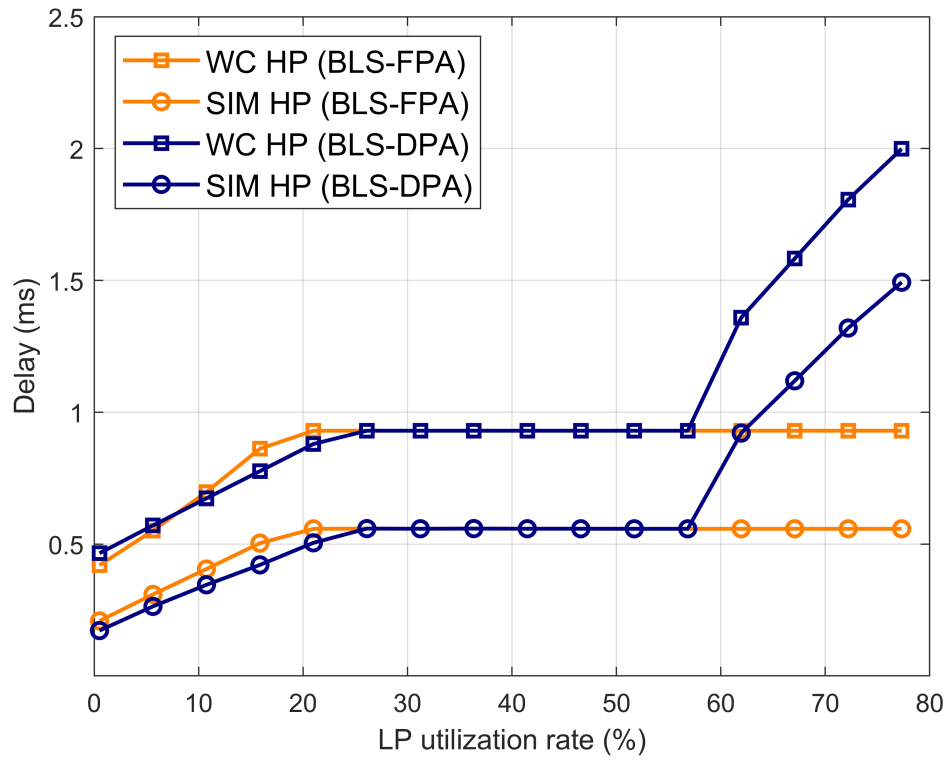


Figure 4.12 The comparison of the worst-case and simulation delay of HP class as UR_{LP} varies (single-hop scenario)

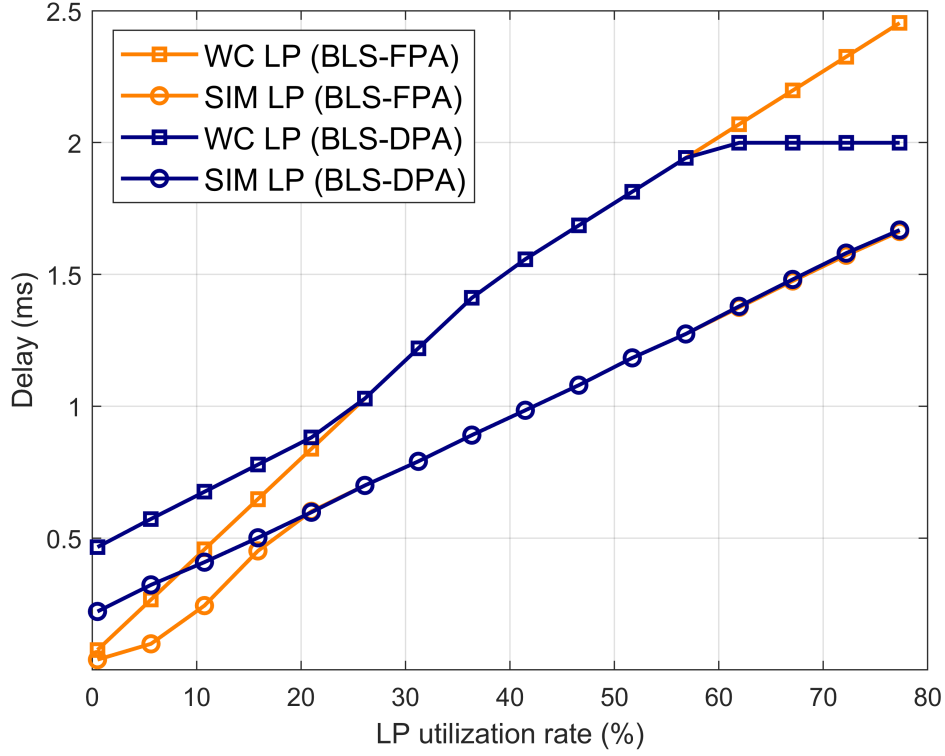


Figure 4.13 The comparison of the worst-case and simulation delay of LP class as UR_{LP} varies (single-hop scenario)

4.2 Performance Evaluation for Multi-Hop Model

In this section, we demonstrate the delay performances of BLS-FPA and BLS-DPA methods on multi-hop network topology by providing numerical experiments and simulations. Note that we use the BLS-FPA method as a benchmark for our proposed priority assignment based BLS-DPA method. Furthermore, we also present the simulation results of these methods to verify the theoretical network calculus results.

For all experiments, the multi-hop system model described in Section 3.2 is considered and the same scenarios in [23] are utilized. Each end system generates an equal number of VL flows for each priority level such that $n_p^{es_1} = n_p^{es_2} = \dots = n_p^{es_{64}}$, where n represents the number of flows and p represents the priority level ($p \in \{HP, LP, BE\}$). Each VL flow originating from a source end system is routed to 16 different destination end systems after crossing two successive switches. The first switch routes the frames in a multi-cast way to transmit the VL flows to two successive switches. The second switch, receiving the VL flows from two previous switches, routes the flows to the destination end systems. Thus each destination end system receives VL flows from 16 source end systems. Fig. 4.14 represents the transmission pattern for two VL flows generated from different source end systems while Fig. 4.15 demonstrates the simulation environment for multi-hop topology.

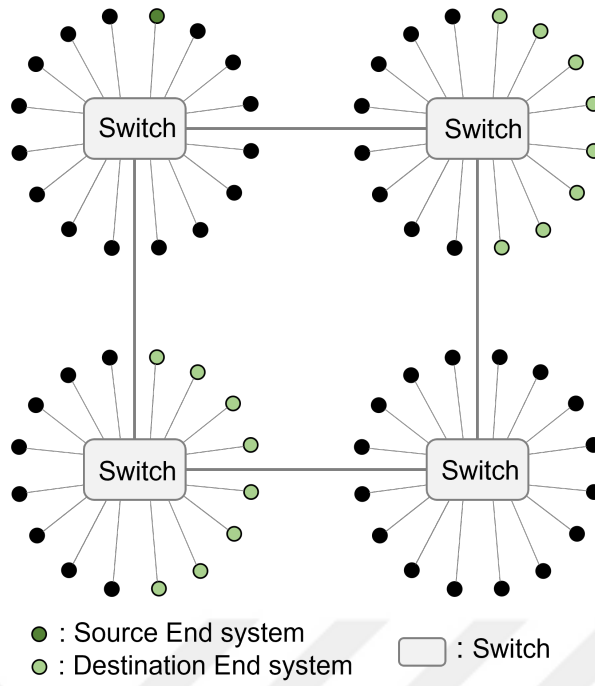


Figure 4.14 Multi-hop scenario

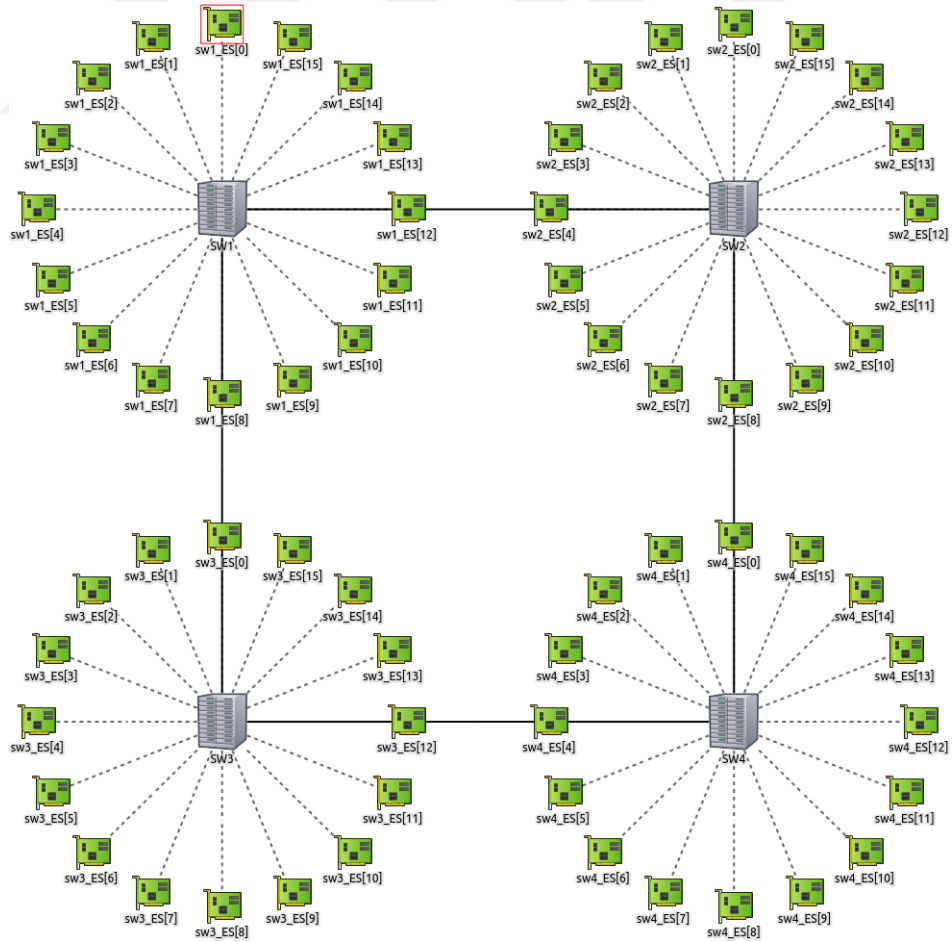


Figure 4.15 OMNeT++ implementation of multi-hop system model

We perform two different experiments. In the first experiment, the utilization rate of the HP class (UR_{HP}) is increased from 0.4% to 44.6% by adding new HP flows, whereas the utilization rate of the LP class (UR_{LP}) remains constant at 20.48%. In the second set of experiments, UR_{LP} is varied from 2.05% to 79.9% by adding new LP flows, while UR_{HP} is held constant at 19.25%. Note that the BE is used to load the system up to the 100% utilization rate. The delay analysis of the BE is not presented since there is no deadline constraint. The same experiment parameters in [23] are utilized. The bandwidth allocation gap (BAG) and the maximum frame size (MFS) are set to 2 ms and 64 bytes for SCT and RC flows, respectively, whereas MFS for BE is 1024 bytes and BAG is 2 ms. The L_M , L_R , and BW parameters are set to 22077, 0, and 0.46, respectively as in [23].

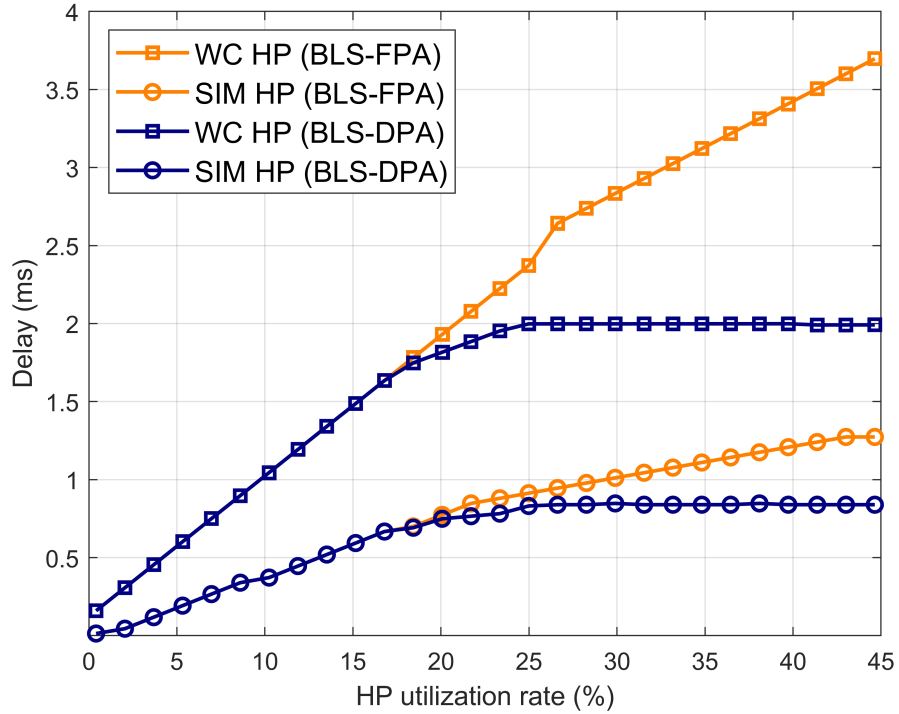


Figure 4.16 The comparison of the worst-case and simulation delay of HP class as UR_{HP} varies (multi-hop scenario)

We report the worst-case (WC) and simulation delays of BLS-FPA and BLS-DPA for all experiments. Fig. 4.16 and Fig. 4.17 demonstrate the effect of the HP utilization rate on the delay results of BLS-FPA and BLS-DPA methods by providing both the worst-case and simulation delay results of HP and LP classes, respectively. For both BLS-FPA and BLS-DPA methods, the delay results of HP and LP classes are the same among themselves until UR_{HP} reaches 16.8%. However, an undesirable behavior is occurred such that the worst-case delay for the HP class is higher than the worst-case delay for the LP class during the HP utilization rate between 16.8% to 25%. This issue is avoided by BLS-DPA method by moving the HP flows (starting from the lowest

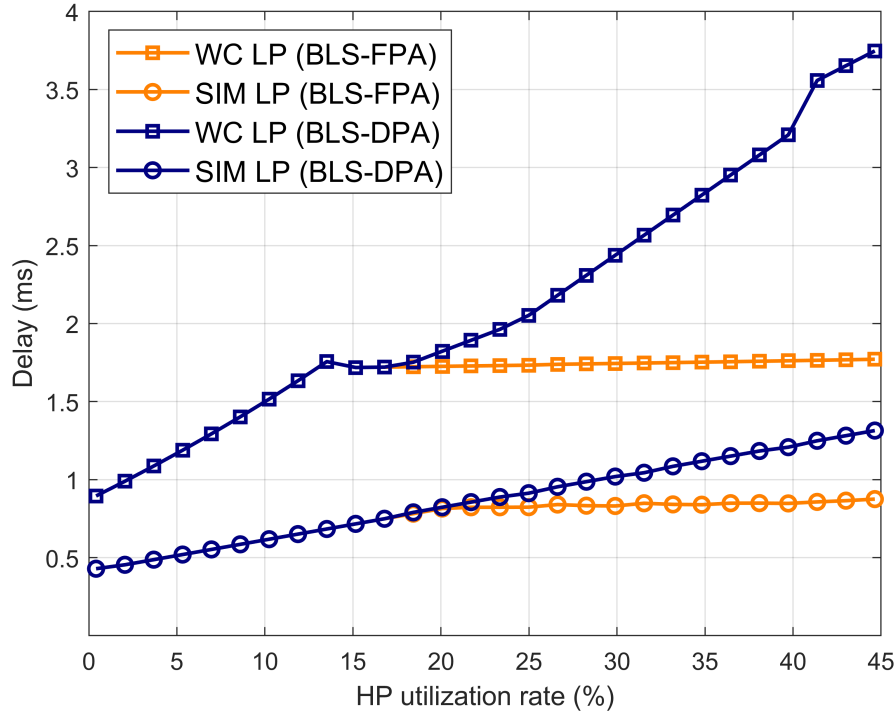


Figure 4.17 The comparison of the worst-case and simulation delay of LP class as UR_{HP} varies (multi-hop scenario)

priority one) to the LP class until the worst-case delay for the HP class is lower than the LP class. After 25%, BLS-DPA not only solves the undesirable behavior but also avoids the worst-case delay to be higher than the delay constraints defined as 2 ms for the HP class. Note that this improvement is achieved by scarifying the worst-case delay of the LP class as shown in Fig. 4.17. Furthermore, regardless of the HP utilization rate, the delay results obtained from simulations are always lower than the theoretical worst-case delay results for both methods, namely, BLS-FPA and BLS-DPA.

Similarly, Fig. 4.18 and Fig. 4.19 illustrate how the change in UR_{LP} affects the worst-case and simulation delay results of HP and LP classes utilizing BLS-FPA and BLS-DPA methods, respectively. Until UR_{HP} reaches 22.5%, BLS-FPA again suffers from the above mentioned undesired issue that the worst-case delay for the HP class is higher than the LP class. BLS-DPA avoids this problem by reassigning HP flows to the LP class starting from the lowest priority one. After this critical point, the worst-case delay for the LP class exceeds the deadline constraint (2 ms) for BLS-FPA. In contrast, the delay constraints for both HP and LP classes are satisfied by the BLS-DPA method that moves some of the HP flows into the LP class. This re-assignment process increases the number of LP flows satisfying the delay constraint. Although this priority re-assignment increases the worst-case delay of the HP class, it does not pose a problem since the HP flows still satisfy the delay requirement. Moreover, the

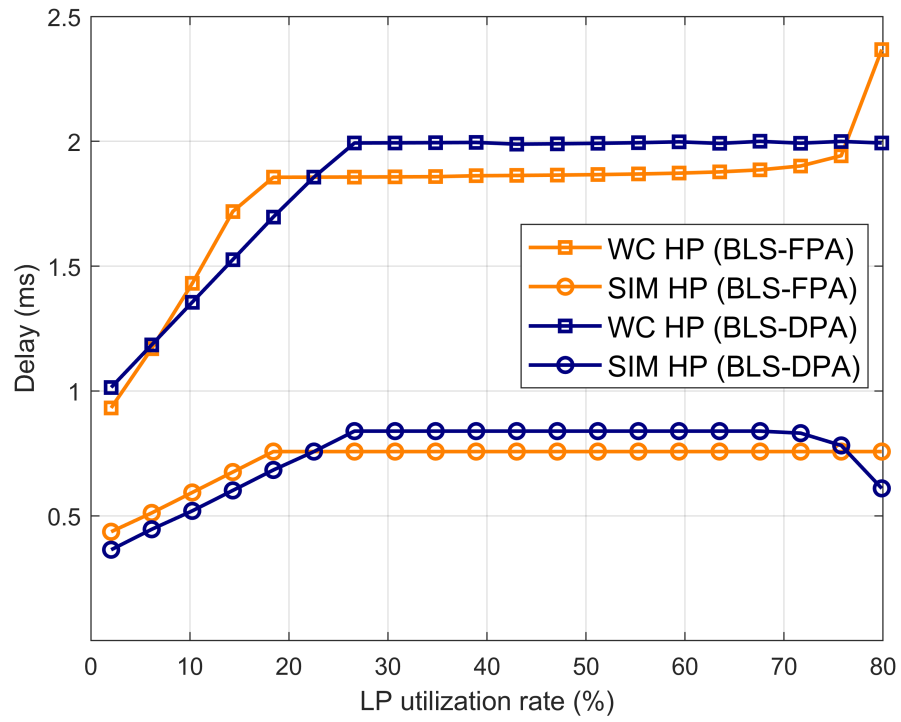


Figure 4.18 The comparison of the worst-case and simulation delay of HP class as UR_{LP} varies (multi-hop scenario)

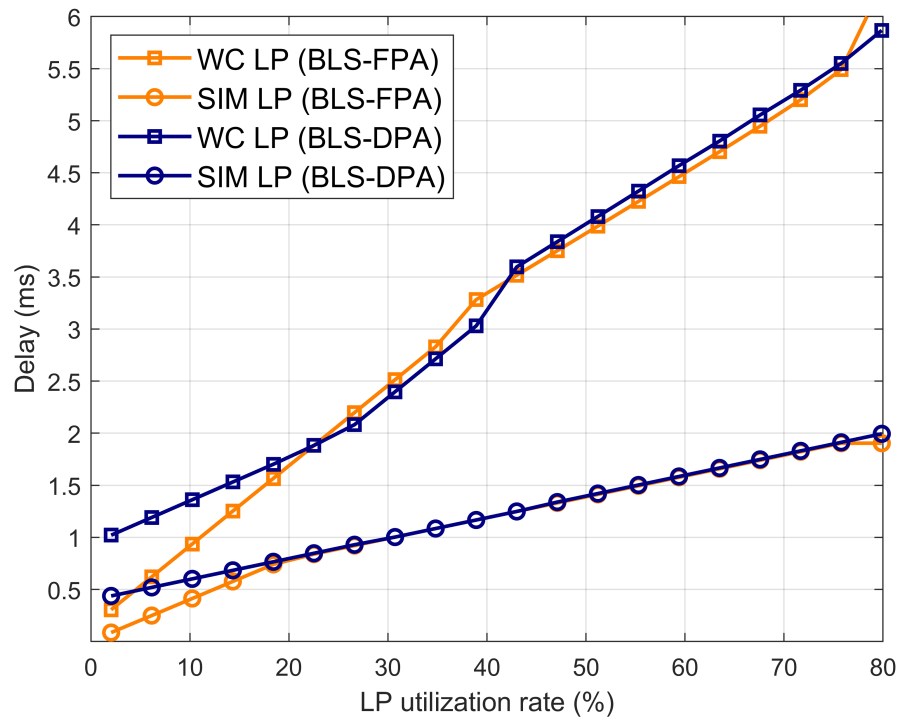


Figure 4.19 The comparison of the worst-case and simulation delay of LP class as UR_{LP} varies (multi-hop scenario)

simulation results have a similar characteristic confirming the theoretical results and are always below the worst delay value.



5

RESULTS AND DISCUSSION

This thesis presented a DPA concept to decide the mapping of ARINC 664 virtual link (VL) flows to the priority classes according to time-varying bandwidth utilization rates. The DPA concept allows us to modify the initial assignment of the VL flows to the priority classes as the class utilization rates change. The priority re-assignment is performed carefully such that the deadline constraints of the HP and LP flows are simultaneously considered. The DPA concept is applied with and without the IEEE TSN BLS by calculating the HP and LP class worst-case delays using the network calculus framework.

The algorithm is analyzed on two case studies, namely single-hop and multi-hop. The single-hop case study is utilized to verify the effectiveness of the algorithm, whereas the multi-hop case study is utilized to evaluate DPA on realistic network topology. The numerical results demonstrate that the DPA method with the IEEE TSN BLS maximizes the bandwidth utilization rate for the HP class first, and the remaining capacity is used to meet the deadline constraints for the LP class. This indicates that the DPA with the IEEE TSN BLS can improve the schedulability and mixed-criticality of ARINC 664 without using any complex time synchronization mechanism. Furthermore, we provide the simulation results for each case study to verify the theoretical network calculus formulation. The simulation results are consistently below the theoretical upper bound delay values for all experiments as well as the graphs obtained from the simulation and theoretical delay results show similar characteristics. Thus, we conclude that the theoretical formulas are consistent with the BLS algorithm implemented in the simulation model.

Future work will investigate a dynamic learning based flow re-assignment algorithm and the use of the proposed algorithm in a network topology where different delay constraints are specified for each VL flow.

REFERENCES

- [1] A. Hughes, A. Awad, "Quantifying performance determinism in virtualized mixed-criticality systems," in *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*, 2019, pp. 181–184. DOI: 10.1109/ISORC.2019.00041.
- [2] *MIL-STD-1553B, military standard: aircraft internal digital time division command response multiplex data bus*. United States Department of Defense, 1978.
- [3] *Aeronautical Radio Incorporated. ARINC specification 429P3-19, Mark 33 digital information transfer system (DITS) - Part 3 - File data transfer techniques*. INC Aronautical Radio, 2009.
- [4] H. Wang, W. Niu, "A review on key technologies of the distributed integrated modular avionics system," *International Journal of Wireless Information Networks*, vol. 25, no. 3, pp. 358–369, 2018.
- [5] C. M. Fuchs, A. S. Schneelee, "The evolution of avionics networks from ARINC 429 to AFDX," *Innovative Internet Technologies and Mobile Communications (IITM), and Aerospace Networks (AN)*, vol. 65, pp. 1551–3203, 2012.
- [6] O. Hamid, S. A. Vaqar, "A comparison of distributed data communications using Ethernet in aircraft," 2020.
- [7] S. Cevher, A. Mumcu, A. Caglan, E. Kurt, M. K. Peker, I. Hokelek, S. Altun, "A fault tolerant software defined networking architecture for integrated modular avionics," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–9. DOI: 10.1109/DASC.2018.8569681.
- [8] G. Chomik, "Application of Ethernet network in avionics systems,"
- [9] K. S. V. S., "The Overview of Avionics Full-Duplex Switched Ethernet," *International Journal of Advanced Engineering Research and Science*, vol. 3, no. 11, Nov. 2016.
- [10] *Time-Triggered Ethernet, SAE AS6802*. SAE International Std, 2011.
- [11] J. Farkas, L. L. Bello, C. Gunther, "Time-sensitive networking standards," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 20–21, 2018.
- [12] *ARINC 664, P7: Avionics Full Duplex Switched Ethernet (AFDX) Network*. INC Aronautical Radio, 2005.
- [13] S. Majumder, J. Nielsen, A. La Cour-Harbo, H. Schiøler, T. Bak, "A real-time on-chip network architecture for mixed criticality aerospace systems," *The Aeronautical Journal*, vol. 123, no. 1269, pp. 1788–1806, 2019. DOI: 10.1017/aer.2019.80.

- [14] T. Gaska, C. Watkin, Y. Chen, "Integrated modular avionics-past, present, and future," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 9, pp. 12–23, 2015.
- [15] E. C. Akpolat, Ö. F. Gemici, M. S. Demir, I. Hokelek, S. Coleri, H. A. Çırpan, "Genetic algorithm based ARINC 664 mixed criticality optimization using network calculus," in *WS02 IEEE ICC 2021 Workshop on Time-sensitive and Deterministic Networking (WS02 ICC'21 Workshop - TsDN)*, Montreal, Canada, Jun. 2021.
- [16] E. Kyriakakis, M. Lund, L. Pezzarossa, J. Sparsø, M. Schoeberl, "A time-predictable open-source TTEthernet end-system," *Journal of Systems Architecture*, vol. 108, p. 101 744, 2020.
- [17] L. L. Bello, W. Steiner, "A perspective on IEEE time-sensitive networking for industrial communication and automation systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019.
- [18] L. Zhao, F. He, E. Li, J. Lu, "Comparison of Time Sensitive Networking (TSN) and TTEthernet," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–7. DOI: 10.1109/DASC.2018.8569454.
- [19] E. Erdinç, "Soft AFDX (avionics full duplex switched ethernet) end system implementation with standard PC and Ethernet card," M.S. thesis, Middle East Technical University, 2010.
- [20] A. Finzi, A. Mifdaoui, F. Frances, E. Lochin, "Incorporating TSN/BLS in AFDX for mixed-criticality applications: Model and timing analysis," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, IEEE, 2018, pp. 1–10.
- [21] A. Finzi, A. Mifdaoui, "Worst-case timing analysis of AFDX networks with multiple TSN/BLS shapers," *IEEE Access*, vol. 8, pp. 106 765–106 784, 2020.
- [22] D. Thiele, R. Ernst, "Formal worst-case timing analysis of Ethernet TSN's burst-limiting shaper," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016, pp. 187–192.
- [23] A. FINZI, A. MIFDAOUI, F. FRANCES, E. LOCHIN, "Network calculus-based timing analysis of AFDX networks with strict priority and TSN/BLS shapers," in *2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES)*, 2018, pp. 1–10. DOI: 10.1109/SIES.2018.8442080.
- [24] M. N. Durceylan, Ö. F. Gemici, İ. Hökelek, H. Aşmer, "Reconfigurable ARINC 664 end system implementation on FPGA," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 2020, pp. 1–4. DOI: 10.1109/SIU49456.2020.9302483.
- [25] M. Boyer, N. Navet, M. Fumey, J. Migge, L. Havet, T. Avionics, "Combining static priority and weighted round-robin like packet scheduling in AFDX for incremental certification and mixed-criticality support," in *5TH European Conference for Aeronautics and Space Sciences (EUCASS)*, 2013.
- [26] J. Yao, J. Wu, Q. Liu, Z. Xiong, G. Zhu, "System-level scheduling of mixed-criticality traffics in avionics networks," *IEEE Access*, vol. 4, pp. 5880–5888, 2016.

- [27] D. TamasSelicean, P. Pop, W. Steiner, "Timing analysis of rate constrained traffic for the TTEthernet communication protocol," in *2015 IEEE 18th International Symposium on Real-Time Distributed Computing*, IEEE, 2015, pp. 119–126.
- [28] M. Abuteir, R. Obermaisser, "Simulation environment for Time-Triggered Ethernet," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, IEEE, 2013, pp. 642–648.
- [29] T. Steinbach, H. D. Kenfack, F. Korf, T. C. Schmidt, "An extension of the OMNeT++ INET framework for simulating real-time Ethernet with high accuracy," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, 2011, pp. 375–382.
- [30] L. Zhao, P. Pop, Q. Li, J. Chen, H. Xiong, "Timing analysis of rate-constrained traffic in TTEthernet using network calculus," *Real-Time Systems*, vol. 53, no. 2, pp. 254–287, 2017.
- [31] W. Steiner, B. Dutertre, "The TTEthernet synchronisation protocols and their formal verification," *International Journal of Critical Computer-Based Systems* 17, vol. 4, no. 3, pp. 280–300, 2013.
- [32] A. Finzi, "Specification and analysis of an extended AFDX with TSN/BLS shapers for mixed-criticality avionics applications," Ph.D. dissertation, Toulouse, ISAE, 2018.
- [33] D. Thiele, R. Ernst, J. Diemer, "Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers," in *2015 IEEE Vehicular Networking Conference (VNC)*, IEEE, 2015, pp. 251–258.
- [34] V. Gavriluț, P. Pop, "Scheduling in Time Sensitive Networks (TSN) for mixed-criticality industrial applications," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, IEEE, 2018, pp. 1–4.
- [35] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, L. L. Bello, "Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support," *Real-Time Systems*, vol. 53, no. 4, pp. 526–577, 2017.
- [36] D. Maxim, Y.-Q. Song, "Delay analysis of AVB traffic in Time-Sensitive Networks (TSN)," in *Proceedings of the 25th International Conference on Real-Time Networks and Systems*, 2017, pp. 18–27.
- [37] S. Brunner, J. Roder, M. Kucera, T. Waas, "Automotive E/E-architecture enhancements by usage of Ethernet TSN," in *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*, IEEE, 2017, pp. 9–13.
- [38] N. Navet, J. Villanueva, J. Migge, M. Boyer, "Insights on the performance and configuration of AVB and TSN in automotive applications," 2017.
- [39] S. Thangamuthu, N. Concer, P. J. Cuijpers, J. J. Lukkien, "Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2015, pp. 55–60.
- [40] J. Specht, S. Samii, "Urgency-based scheduler for time-sensitive switched Ethernet networks," in *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, IEEE, 2016, pp. 75–85.

- [41] A. Arestova, K.-S. J. Hielscher, R. German, “Simulative evaluation of the TSN mechanisms Time-Aware Shaper and frame preemption and their suitability for industrial use cases,” in *2021 IFIP Networking Conference (IFIP Networking)*, IEEE, 2021, pp. 1–6.
- [42] M. E. Yabacioğlu, M. S. Demir, S. N. Engin, *et al.*, “FPGA implementation of Precision Time Protocol and its performance analysis under traffic load,” in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2021, pp. 1–4.
- [43] S. Chouksey, H. S. Satheesh, J. Åkerberg, “An experimental study of TSN-nonTSN coexistence,” in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2021, pp. 1577–1584.
- [44] Z. Zhou, J. Lee, M. S. Berger, S. Park, Y. Yan, “Simulating TSN traffic scheduling and shaping for future automotive Ethernet,” *Journal of Communications and Networks*, vol. 23, no. 1, pp. 53–62, 2021.
- [45] A. Gogolev, P. Bauer, “A simpler TSN? traffic scheduling vs. preemption.,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, vol. 1, 2020, pp. 183–189.
- [46] D. Ergenç, M. Fischer, “On the reliability of IEEE 802.1 CB FRER,” in *IEEE INFO-COM 2021-IEEE Conference on Computer Communications*, IEEE, 2021, pp. 1–10.
- [47] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, H. ElBakoury, “Reconfiguration algorithms for high precision communications in time sensitive networks: Time-aware shaper configuration with IEEE 802.1 Qcc (extended version),” *arXiv preprint arXiv:1906.11596*, 2019.
- [48] F. Luo, B. Wang, Z. Fang, Z. Yang, Y. Jiang, “Security analysis of the TSN backbone architecture and anomaly detection system design based on IEEE 802.1 Qci,” *Security and Communication Networks*, vol. 2021, 2021.
- [49] W. Quan, J. Yan, X. Jiang, Z. Sun, “On-line traffic scheduling optimization in IEEE 802.1 Qch based Time-Sensitive Networks,” in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, 2020, pp. 369–376.
- [50] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, H. Elbakoury, “Performance comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS),” *IEEE Access*, vol. 7, pp. 44 165–44 181, 2019.
- [51] E. Neretin, E. Lunev, A. Ivanov, A. Budkov, “Application of distributed integrated modular avionics concept for perspective aircraft equipment control systems,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1353, 2019, p. 012 005.
- [52] J. Fernández, H. Pérez, J. J. Gutiérrez, M. G. Harbour, “AFDX emulator for an ARINC-based training platform,” in *Ada-Europe International Conference on Reliable Software Technologies*, Springer, 2015, pp. 212–227.
- [53] J. Vila-Carbó, J. Tur-Massanet, E. Hernández-Orallo, “Analysis of switched Ethernet for real-time transmission,” in *Factory Automation*, IntechOpen, 2010.

- [54] M. E. Altuntas, M. Seker, P. Kışla, E. C. Akpolat, I. Hökelek, M. S. Demir, E. Akdoğan, “Testing deterministic avionics networks using orthogonal arrays,”
- [55] A. Bouillard, M. Boyer, E. Le Corronc, *Deterministic Network Calculus: From Theory to Practical Implementation*. John Wiley & Sons, 2018.
- [56] F. He, L. Zhao, E. Li, “Impact analysis of flow shaping in Ethernet-AVB/TSN and AFDX from network calculus and simulation perspective,” *Sensors*, vol. 17, no. 5, p. 1181, 2017.
- [57] A. Finzi, A. Mifdaoui, F. Frances, E. Lochin, “Performance enhancement of extended AFDX via bandwidth reservation for TSN/BLS shapers,” *ACM SIGBED Review*, vol. 16, no. 1, pp. 21–26, 2019.
- [58] S. Kaya, B. Gül, M. S. Demir, I. Hökelek, S. Coleri, M. Garip, H. Üvet, “Delay optimization for switchless ARINC 664 mesh networks with cyclic dependencies,” in *2021 IEEE Latin-American Conference on Communications (LATINCOM)*, IEEE, 2021, pp. 1–6.
- [59] B. Gül, M. Garip, *et al.*, “Latency analysis of switchless ARINC 664 using network calculus,” in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2021, pp. 1–4.
- [60] H. Bauer, J.-L. Scharbarg, C. Fraboul, “Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach,” *IEEE Transactions on Industrial informatics*, vol. 6, no. 4, pp. 521–533, 2010.
- [61] E. C. Akpolat, M. Şeker, S. Cevher, E. Şapla, S. Ö. Ata, “An OMNeT++ simulation for performance analysis of ARINC 664 P7 avionics data network,” in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2020, pp. 1–4.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. M. Yeniaydın, Ö. F. Gemici, M. S. Demir, I. Hökelek, S. Coleri, U. Tureli, “Priority re-assignment for improving schedulability and mixed-criticality of arinc 664,” in 2021 IFIP Networking Conference (IFIP Networking), IEEE, 2021, pp. 1–6. 52

