

An Independent Timing Analysis for Credit-Based Shaping in Ethernet TSN

Citation for published version (APA):

Cao, J. (2023). *An Independent Timing Analysis for Credit-Based Shaping in Ethernet TSN*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Eindhoven University of Technology.

Document status and date:

Published: 06/06/2023

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

An Independent Timing Analysis for Credit-Based Shaping in Ethernet TSN

~~best-case scenario / het slechtste scenario / het beste scenario / 最差情况 / 最好情况 / el peor de los casos / el mejor de los casos / بدترین سناریو / بهترین سناریو / o cenário de pior caso / o cenário de melhor caso / the worst-case scenario / the best-case scenario / het slechtste scenario / het beste scenario / 最差情况 / 最好情况 / el peor de los casos / el mejor de los casos / بدترین سناریو / بهترین سناریو / o cenário de pior caso / o cenário de melhor caso / the worst-case scenario / the best-case scenario / het slechtste scenario / het beste scenario / 最差情况 / 最好情况 / el peor de los casos / el mejor de los casos / سناریو / o cenário de pior caso / o cenário de melhor caso / best-case scenario / het slechtste scenario / peor de los casos / el mejor~~

peor de los casos / el mejor caso / o
/ het slechtste scenario / het beste scenario
o cenário de pior caso / o cenário de melhor caso / the
the best-case scenario / het slechtste scenario / het beste
el peor de los casos / el mejor de los casos / بدترین سناریو بهترین
worst-case scenario / the
/ 最差情况 / 最好情况 / el
pior caso / o
het slechtste scenario / het beste scenario
o cenário de pior caso / o cenário de melhor caso / the
the best-case scenario / het slechtste scenario / het beste scenario
scenario / 最差情况 / 最好情况 / el peor de los casos / el mejor de los
best-case scenario / the best-case scenario / het slechtste scenario / het beste scenario
peor de los casos / el mejor de los casos / بدترین سناریو بهترین / o cenário de pior caso / o cenário de melhor caso / the best-case scenario / het
ste scenario / het beste scenario / 最差情况 / 最好情况 / el peor de los casos / el mejor de los casos / o cenário de pior caso / o cenário de melhor caso
los casos / o cenário de pior caso / o cenário de melhor caso / the best-case scenario / het slechtste scenario / het beste scenario
scenario / el peor de los casos / o cenário de pior caso / o cenário de melhor caso / the best-case scenario / het slechtste scenario / het beste scenario

Jingyue Cao

An Independent Timing Analysis for Credit-Based Shaping in Ethernet TSN

Jingyue Cao



The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics). This research is a project (12697) under rCPS program: Control based on Data-Intensive Sensing, which was supported by the Dutch Technology Foundation STW with industrial partners FEI (later acquired by Thermo Fisher Scientific) and Technolution.

IPA dissertation series 2023-04

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-5756-1

Cover design: Jiangxue Xu. <https://jiangxue-xu.com>

Printed by Gildeprint, Enschede

©Jingyue Cao, 2023.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronically, mechanically, photocopying, recording or otherwise, without prior permission of the author.

An Independent Timing Analysis for Credit-Based Shaping in Ethernet TSN

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische
Universiteit Eindhoven, op gezag van de rector magnificus
prof.dr. S.K. Lenaerts, voor een commissie aangewezen
door het College voor Promoties, in het openbaar te
verdedigen op dinsdag 6 juni 2023 om 13:30 uur

door

Jingyue Cao

geboren te Ningbo, China

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

Voorzitter:	prof.dr. M.G.J. van den Brand
Promotor:	prof.dr. J.J. Lukkien
Copromotor:	dr.ir. P.J.L. Cuijpers
Leden:	prof.dr.ir. T. Basten
	dr. M. Ashjaei (Mälardalen University)
	dr. L. Almeida (University of Porto)
	prof.dr. M. Behnam (Mälardalen University)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Acknowledgments

As the sun sets on my mother's 66th birthday in the Chinese lunar age, our family gathers to celebrate this special moment. After a prolonged separation due to the Covid pandemic, it is a cherished occasion to reunite and share the joy of the day. Following the celebration, I sit down to write the acknowledgments section of my doctoral dissertation, which is scheduled to be defended on June 6th of this year. At this final phase, I find myself experiencing a mixture of emotions, but the feeling of relaxation and inner peace resonates most strongly. Then, I realize the interesting coincidence that so many 6s are involved – my mother's age and the date of my defense. In Chinese culture, the number 6 is considered a symbol of success, smoothness, and good fortune. I hope that my defense goes well and that my mother and family have a wonderful year ahead.

Reflecting on my Ph.D. journey, I acknowledge the significant challenges it has entailed. The project I engaged in was considerably different from my previous research in terms of subject and methods. If given another opportunity, I might opt for something other than such a project, especially when it involved so many mathematics proofs. Despite having a clear objective in mind, the process of achieving proper proof was time-consuming and filled with obstacles. I often encountered moments of frustration and feeling stuck, which required me to revise definitions from scratch to ensure solid proof. Nevertheless, it is important to recognize the role that these formal methodologies and mathematical proofs have played in providing valuable insights, solidifying our findings, and steering new directions in our research.

Additionally, after giving birth to my son, I encountered an unexpected challenge: the Covid pandemic. The restrictions imposed during this global health crisis prevented my parents and parents-in-law from visiting, and daycare options were not available. Furthermore, my husband's demanding onsite job left me without any daytime assistance. Consequently, I had no alternative but to pause my work on the dissertation. The unpredictable nature of the pandemic made it exceptionally difficult to plan for the worst-case scenario and navigate the completion of my Ph.D. dissertation.

Even after resuming my dissertation, I still struggled with having limited time to dedicate consistently. Taking care of my energetic son during a challenging period of getting a place in daycare and preschool, coupled with managing my new job responsibilities, further complicated the situation. It was far from easy to pick up where I had left off in my research now and then, as I lacked extended blocks of time to make rapid progress. Instead, I could only work at a pace I would describe as my best effort.

However, despite the challenges, I have never considered giving up or quitting because I have had a strong support system throughout this journey. For this, I would like to express my sincere gratitude to my promoter prof.dr. Johan J. Lukkien, co-promoter dr.ir. Pieter J.L. Cuijpers, and my mentor dr.ir. Reinder Bril. Their comprehensive support and mentorship have played a vital role in shaping my research direction and methodology.

First and foremost, I would like to express my deepest gratitude to my daily supervisor, Pieter, for his unwavering guidance throughout my entire Ph.D. journey. Collaborating with him from the very beginning has been a pleasure because he radiates such positive and cheerful energy, and his sense of humor is infectious. His ability to think divergent has often brought fresh and innovative perspectives. Engaging in discussions with him has become a source of inspiration for my research. Moreover, Pieter's commitment to scientific rigor has had a significant impact on the acceptance of our work. Once my research met his standards, objections during paper submissions were seldom encountered. Furthermore, his dedication to his students is truly admirable, as he consistently provided support and assistance whenever it was needed. In particular, during the final phase of finalizing my dissertation, Pieter generously spared time on a weekly basis to review and provide valuable feedback on my proofs and storyline. His constant encouragement for each small step I took was truly motivating. I credit his support as one of the critical factors that helped me reach a lower bound of time for completing my dissertation.

Next, I extend my heartfelt appreciation to my promoter, Johan, for granting me the opportunity to pursue my Ph.D. At first glance, I held the impression that he would be a strict and serious person to work with, but I quickly realized that this was not the case. Despite his demanding role as the department dean, Johan devoted a considerable amount of time to his Ph.D. student. Johan not only effectively steers the research process but also shows genuine enthusiasm for delving into the technical details. His ability to absorb information, ask insightful questions, and provide constructive feedback consistently amazes me. I am especially thankful for Johan's kind words of recognition regarding my work. His acknowledgment was a powerful source of motivation for me to complete my work during the

final phase.

Lastly, I express my sincere gratitude to Reinder for his invaluable contributions throughout my Ph.D. journey. Although not officially my co-promoter, Reinder has been an integral part of my research process, acting as my *de facto* second co-promoter. His concrete and to-the-point advice has consistently guided me towards achieving clarity and depth in my research. He has always been ready to provide timely advice and has shown a remarkable ability to notice even the most minor details in my papers and presentations. I am genuinely thankful for his guidance and expertise, which have greatly enhanced the quality of my work.

In addition to my direct supervisors, I also want to convey my sincere thanks to the members of my doctoral committee, prof.dr.ir. Twan Basten, dr. Mohammad Ashjaei, dr. Luis Almeida, and prof.dr. Moris Behnam for their valuable time, expertise, and constructive feedback, which have enhanced the quality of my dissertation.

To my colleagues Martijn van den Heuvel, Milosh Stolikj, Aleksandra Kuzmanovska, and Amir Soltani Nezhad, I would like to express my sincere gratitude. From the moment I joined the group, they embraced me with open arms and made me feel welcome. I also want to thank the latecomers Leila Fatmasari Rahman, Leo Hatvani, Redegeld Jeroen, Tianyu Bi, Nan Yang, and Hamid Hassani. Despite working on different projects and not having the opportunity to collaborate in a professional setting, we formed a close-knit group. We often had lunch together, went for walks, and had dedicated lunch meetings to discuss our research problems. These interactions provided us with valuable insights and created a warm and supportive environment.

A special mention of appreciation goes to our excellent secretaries Anjolein Gouma and Jolande Matthijsse. They have consistently displayed kindness, approachability, and efficiency in handling any office-related questions or concerns.

I had limited opportunities to supervise students during my Ph.D. time; nevertheless, I consider myself extremely fortunate to have the privilege of mentoring two outstanding master's students. The first student, Nan Yang, volunteered to work with me on a toy project during her summer vacation. Despite being a bit shy initially, her diligence and commitment were already quite impressive. The second student, Hector Joao Rivera Verduzco, demonstrated exceptional academic abilities in tackling complex problems. It has been an enjoyable experience to co-supervise him and collaborate on co-authoring work.

I am deeply grateful for my companion in the STW rCPS program, Robinson Medina, a warm and highly-skilled Ph.D. candidate in embedded systems. Our professional collaborations and personal interactions have been both delightful. I enjoyed our discussions on control theory and MATLAB Simulink, as well as attending STW and user committee meetings together. I also appreciate another companion Garbí Singla Lezcano for fostering a vibrant atmosphere during our intern days in FEI.

Speaking of FEI, later acquired by Thermo Fisher Scientific, I express my heartfelt gratitude for the invaluable opportunity it provided me to delve into the practical applications of electron microscopy. This opportunity not only expanded my horizons of this advanced technology but also connected me with a network of brilliant and compassionate individuals to collaborate and learn with. I sincerely thank Bart Janssen and Jeroen De Boeij for their instrumental role in organizing this incredible experience.

It was during that time that I had the privilege of forming lasting friendships with Lingbo Yu, Yuchen Deng, and Qi Liu. Among them, I was fortunate to meet Lingbo Yu, who has become one of my closest friends. It was remarkable that despite coming from the same hometown, attending the same university, and pursuing the same major, we had never met before FEI. After I relocated to Utrecht, Lingbo graciously shared her room, and we enjoyed cooking dinners together. Our friendship was filled with countless cherished moments where we discussed technology, research, women in technology, and our personal lives. Lingbo's expertise and passion in cryo-electron microscopy are incredibly enlightening, and our conversations during that time hold a special place in my heart.

I would also like to extend my gratitude to Reinder and Pieter for their invaluable assistance in establishing a collaboration with the Networked and Embedded Systems department at Mälardalen University in Västerås, Sweden. I am deeply thankful for Mohammad Ashjaei and prof.dr. Moris Behnam for giving me the opportunity to collaborate on a joint paper. A special thank you goes to prof.dr. Saad Mubeen for generously dedicating his time to discuss the proof I was working on.

Additionally, I appreciate Mohammad and my temporary office mate, Leo, for their warm hospitality. Their kindness made me feel like a part of their big family and allowed me to immerse myself in the social life of their department. It not only facilitated our joint work seamlessly but also added a touch of joy and fulfillment to my stay in Västerås.

During the final stages of my dissertation and defense preparation, I am immensely grateful to Nan Yang for her support. Her willingness to share her own experience in completing her dissertation has been truly in-

valuable. Moreover, I am grateful to Nan for introducing me to the talented cover designer, Jiangxue Xu. Her ability to effortlessly grasp my ideas, meet deadlines and dedicate herself to meeting my requirements resulted in a flawless final cover design. I consider myself fortunate to have had Jiangxue's expertise in making the final step of my dissertation truly perfect.

Besides my supervisors, colleagues, and friends, I also want to express my heartfelt thanks to my family. I am deeply grateful to my beloved son, Muzhi Guo, for choosing me as his mother. His presence in my life has brought both exhaustion and immeasurable joy. While it is undeniable that caring for him has posed challenges and negatively impacted my progress in my dissertation, I have realized that he has become a remarkable source of motivation, pushing me to persist in the face of obstacles. I strive to be the best version of myself, not only for my own personal growth but also as a role model for him.

I would like to thank my father, Weining Cao, and my mother, Longdi Yang, for their devotion as parents. From a young age, they instilled in me the core values of independence and hard work. In a society that often encourages girls to settle for an easier life, my parents have been a constant source of support, allowing me to pursue higher education and study abroad. I am truly grateful for their openness and flexibility.

Furthermore, I am immensely thankful for their presence by my side in the Netherlands during the final phase of my dissertation and defense preparations. Without their unconditional help, it would have been impossible for me to balance my work responsibilities and care for my son, let alone find the necessary time and space to refine my dissertation. As their daughter, I feel incredibly fortunate and blessed.

I want to give a special thanks to my maternal grandmother. Seeing you healthy and strong during our visit to Ningbo after the pandemic made me feel incredibly lucky. I am also deeply grateful for the love and guidance I received from my maternal grandfather and paternal grandmother, who passed away while I was doing my doctoral studies. I strive to live a life that always makes them proud.

Last but not least, I would like to express my gratitude to my husband, Zong Guo, for being my constant support throughout my Ph.D. journey and in every aspect of my life. Thank him for being the rock behind my back.

In the end, I sincerely thank all the individuals who have been a part of my Ph.D. journey, whether mentioned in the acknowledgments or not. Your presence and support have made this an incredibly memorable and

transformative experience in my life. Just as the network traffic I dealt with has different priorities, it is now time for me to shift my priorities and embark on a new chapter.

Jingyue Cao
09-05-2023
Utrecht

Contents

Acknowledgments	i
1 Introduction	1
1.1 Background	1
1.2 Problem Statement	2
1.3 The Issue with Existing Works	3
1.4 Contribution	4
1.4.1 Relative Worst-Case Response Time	4
1.4.2 Relative Best-Case Response Time for Bursty Traffic .	5
1.4.3 Minimum Bandwidth Reservation Strategy	5
1.5 Outline	6
2 Ethernet TSN and Existing Response Time Analyses	7
2.1 Ethernet TSN	8
2.1.1 Time Synchronization	10
2.1.2 Bounded Latency and Zero Congestion Loss	10
2.1.3 Reliability	12
2.1.4 Traffic Management	13
2.2 Existing Works on Response Time Analysis	14
3 System Model	17
3.1 Mechanics of Credit-Based Shaping	17
3.2 Notation	20
3.3 Formalization of Shaping Rules	22
4 Eligible Intervals	27
4.1 Eligible Intervals	28
4.2 Shaping an Uninterfered Execution	29
4.3 The Start of an Eligible Interval	31
4.4 Credit Represents Delay	34
5 Relative Worst-Case Response Time Analysis	37

5.1	Calculating the Minimum Credit	38
5.2	Bounding the Maximum Relative Delay	41
5.3	Consistency with Previous Works	46
5.3.1	Only Low-Priority Interference	46
5.3.2	Only One High-Priority Interference	47
5.3.3	One High-Priority Combined with Low-Priority Interference	48
5.4	A Concrete Example of Multiple High-Priority Interference	49
5.5	Conclusion	51
6	Tightness of Our Approach	53
6.1	Tightness for a Single High-Priority Interference	53
6.2	Examples Why Tightness Is Not Always Guaranteed for Multiple High-Priority Interference	55
6.3	Tightness for Multiple High-Priority Interference	57
6.4	Conclusion	61
7	Comparison with Earlier Works	63
7.1	Worst-Case Response Time Analysis in Uninterfered Schedules	64
7.2	Illustrative Example: Single High-Priority Interference	69
7.3	Applying Existing Methods to the Same Example	70
7.4	Exploring Pessimism	72
7.5	Adding Information	76
7.6	Conclusion	77
8	Relative Best-Case Response Time Analysis	79
8.1	Introduction	79
8.2	Burstiness	80
8.3	Relative Best-Case Response Time	81
8.4	Conclusion	85
9	Reservation Strategy	87
9.1	Introduction	87
9.2	Related Works on Network Bandwidth Reservation	88
9.3	Bandwidth Reservation Analysis	89
9.3.1	Bandwidth Reservation for the High-Priority Stream	90
9.3.2	Bandwidth Reservation for the Medium-Priority Stream	92
9.4	Experiments	94
9.4.1	Identical Sources	95
9.4.2	Non-identical Sources	100
9.5	Discussion	101
9.5.1	Utilization Constraint vs. Deadline Constraint	101
9.5.2	Merits	102

9.5.3	Demerits	102
9.6	Conclusion and Future Work	103
10	Conclusions and Future Work	105
10.1	Discussion of Main Results	105
10.2	Limitations and Future Work	107
	Bibliography	111
	Summary	115
	Curriculum Vitae	117

Chapter 1

Introduction

1.1 Background

There is a growing number of connected components present in sophisticated cyber-physical systems, e.g., in the automotive industry, high-tech systems, and industrial automation. These systems are networked and have stringent requirements in terms of communication, time, memory, processing power, etc. Those components are of a heterogeneous nature, in the sense that some have hard real-time requirements with low bandwidth demands on the communication network, while others merely require soft real-time guarantees with high bandwidth demands or have no specified requirements at all. For instance, in an electron microscope, we use images or other sensor data to estimate the tiny motion of the specimen and then compensate for it by adjusting the beam and the stage to achieve automated drift compensation. Large volumes of images and other sensor data need to be sent over a network as quickly as feasible for further processing. Then the resultant small-size control information needs to be sent to the actuators within a stringent timing demand. Meanwhile, the operator must be able to inspect the image on display without experiencing an excessive amount of delay. Whether a single network backbone can ensure the large-volume image data is transmitted in time while the small-size control data must fulfill strict timing demands is a challenge that a modern industrial network should cope with.

To face the consequences of this heterogeneity, Ethernet is currently being considered as a possible solution to this problem, as it provides high bandwidth as part of a widely accepted standard. Since Ethernet was originally intended for best-effort transmission, it can not provide real-time guarantees; however, the development of the Ethernet AVB standard (from

the Audio/Video Bridging task group IEEE (2005)), enables the use of Ethernet for transporting high-volume data with latency guarantees. This standard enables prioritized data transmission and relies on traffic shaping techniques, and in particular on the standardized Credit-Based Shaper (see IEEE 802.1Q-2014 IEEE (2014)) to be able to guarantee required latency and throughput for audio/video traffic while preventing starvation of best-effort traffic. In 2012, the task group was renamed to the Time Sensitive Networking (TSN) task group IEEE (2012), with the aim of supporting time-sensitive traffic by introducing time-triggered transmission on top of other traffic classes. This latter enhancement of scheduled traffic (see IEEE 802.1Qbv IEEE (2016)) is not considered in this dissertation as we only focus on Credit-Based Shaping.

To delve into further details, the transmission of frames through an Ethernet AVB Switch's output port is scheduled in a prioritized fashion. Frames from several input ports are distributed over (FIFO) output queues depending on their corresponding priority classes. Because the initial purpose of Ethernet AVB is to serve audio and video traffic in addition to best-effort traffic, priorities are often restricted to three: audio takes precedence over video, and video gets priority over Best-Effort traffic. Within audio and video traffic, a stream of frames is shaped according to the Credit-Based Shaping algorithm. Unlike non-idling servers, which permit transmission anytime network bandwidth is available, Credit-Based Shaping acts as an idling server, only allowing transmission when its credit, in addition to available bandwidth, complies with the transmission rules. As a result, the transmission for those classes is constrained to a fraction of the available bandwidth, and thus provide latency guarantees for the low priorities as well.

1.2 Problem Statement

In this Ethernet-based network, we are interested in the timing analysis for a specific traffic stream of interest to move across the network, primarily in the worst-case and also potentially in best-case scenarios. Bounds for worst-case transmission time are crucial in general; for instance, if your car's rear-view camera fails to deliver an image within a specified amount of time, you may end up in a collision. However, in other circumstances, the best-case response time has a significant influence as well. An airbag in the car provides the highest level of safety in the event of an automobile accident when it is not inflated too early or too late.

For a correct timing analysis, we must prove that the worst-case and best-case bounds we find are safe, meaning that all scenarios that are possi-

ble within a given traffic model will fall within the analyzed ranges. On top of this, we recognize the importance of two other factors in a good timing analysis: tightness and independence.

Tightness describes how well a timing analysis method exploits the information provided to it; in other words, how close the estimated bounds are to what is theoretically possible within the given assumptions made in the traffic model. An analysis is considered tight if, within these assumptions, a scenario can be perceived that achieves the predicted bounds arbitrarily closely. Underestimation is seen as providing unsafe bounds, whereas overestimation results in pessimism. Note that a tight analysis (relative to the traffic model) may still be pessimistic relative to reality when the actual traffic turns out to be more restricted than what is allowed by the traffic model. Although it is not always possible to obtain tight bounds given all possible shaping strategies and traffic models, it is common sense that we would desire as tight estimates as possible.

Independence means that a timing analysis is not affected by changes in the environment of the system. This can partly be done by making as few assumptions about the environment as possible. However, there is a trade-off to be considered. On the one hand, the more information we have about environmental traffic in our traffic model, the more accurate an estimate of the timing should be possible in theory. On the other hand, any changes in this information would render a previous analysis invalid, making the analysis less independent. Independence is important in the design of complex systems, where developers would like to consider the behavior of third-party components as environmental. Changes in those components, or even misbehavior, may then be seen as violations of the assumptions on the environment. An independent analysis makes the system design robust against this.

1.3 The Issue with Existing Works

We have just explained what, in our opinion, a good timing analysis should accomplish. Looking at the existing works, we see that the chosen (classical) method of busy period analysis has some drawbacks considering tightness and independence.

The busy period analysis was originally developed for non-idling servers, which means that network bandwidth is being used whenever there is something to transmit. Credit-Based Shapers do have idling behavior whenever credit is negative. The works of Diemer et al. (2012b), Axer et al. (2014) and Bordoloi et al. (2014), deal with this implicitly by considering idle time

as yet another source of interference, but as pointed out in Bordoloi et al. (2014), this leads to an overestimation of response times. For example, when interfering high-priority traffic is transmitted during an idling period, the interference time is counted twice. The authors attempt to compensate for this pessimism, but our further study indicates that their attempt was not entirely successful. A different approach is needed to obtain an analysis that is tight.

Regarding independence, another disadvantage of using busy period analysis is that the analysis requires detailed traffic models of all sources of interference. Typically, a restrictive model such as periodic or sporadic traffic model is chosen for this interference. Thus, the estimated bound is highly dependent on the selected model and parameters.

In this dissertation, we propose an alternative to the use of busy period analysis, which does allow us to reach independent and (under certain conditions) tight estimates.

1.4 Contribution

The contribution of this dissertation consists of three major parts. Firstly, we propose an independent timing analysis to obtain a relative worst-case response time for traffic under Credit-Based Shaping; secondly, we provide an independent timing analysis for a burst of traffic to obtain a relative best-case response time; thirdly, we determine a minimum bandwidth reservation to ensure the schedulability of network traffic, based on this independent timing analysis.

1.4.1 Relative Worst-Case Response Time

The first and most significant contribution is the introduction of a novel, independent method for performing timing analyses. This new method only requires us to know the parameters of the shapers for other priorities (inter-priority interference), and we ignore the detailed traffic model of interference. Despite the fact that we still need to deal with the traffic model of streams that have the same priority as the one of interest, this results in an initial separation of design concerns and adds more robustness to the timing analysis.

This independent method is realized by introducing Eligible Intervals. Within an Eligible Interval for a certain priority, there exists a pending load of that priority either eligible for transmission or eligible in transmission. This takes the idling nature of Credit-Based Shaping into account.

After analyzing the properties of Eligible Intervals for Credit-Based Shaping, we are able to relate the properties of the interfered schedules to those of uninterfered ones to obtain relative delays. Furthermore, with the assumption of knowing interference only at the shaper level (enforced by the Ethernet TSN standard), we manage to determine the upper bound for the worst-case relative delay given a generic interference consisting of multiple high-priorities under Credit-Based Shaping and multiple low-priorities. As a result, we break down the problem to determine the absolute worst-case response time into two steps: first, we obtain the worst-case response time in an uninterfered schedule, and then, we add the relative worst-case delay to this.

We provide rigorous proof to ensure the upper bounds are safe, followed by a further investigation of the conditions under which this independent analysis is tight.

Comparing our result to the literature we find that, although the existing works use a more restrictive model of interference and in principle could give better results, our estimates turn out to be less pessimistic in most cases.

1.4.2 Relative Best-Case Response Time for Bursty Traffic

Secondly, since best-case response time also plays a significant role, we extend this independent timing analysis to estimate the relative best-case response time, but only for bursty traffic under Credit-Based Shaping. An essential and surprising finding is that adding interference may contribute to an earlier transmission rather than a delay in the start of frame transmissions, resulting in a negative relative best-case response time.

1.4.3 Minimum Bandwidth Reservation Strategy

Thirdly, in addition to the formerly introduced worst-case and best-case response time analysis, it is a difficult challenge to allocate network bandwidth properly and systematically for a distributed data-intensive embedded system. We use this independent timing analysis to serve as the basis for determining a minimum bandwidth reservation for Credit-Based Shapers. While it is mandatory to set the bandwidth reservation at least to the expected utilization, we find that reserving bandwidth based on the utilization of that priority is not always sufficient. More bandwidth may be required to make sure deadlines are met. We conduct a set of comparative experiments and demonstrate an improvement in the bandwidth reservation efficiency compared to the state-of-art work of Ashjaei et al. (2017).

1.5 Outline

This dissertation is comprised of 10 chapters in total. After this introduction, the overview of the prior literature is covered in Chapter 2. Chapter 3 formally describes mathematical notations and relationships that characterize the mechanics of Credit-Based Shaping. Chapter 4 presents the definition of Eligible Interval, the properties of the uninterfered schedules, and the fundamental theorem that links interfered schedules with uninterfered schedules.

Chapter 5 covers our main contribution, as it presents how to use the independent timing analysis to estimate the relative worst-case response time for a generic set-up of interference consisting of multiple high priorities under Credit-Based Shaping and multiple low priorities. Chapter 6 further explores the tightness of the independent timing analysis and Chapter 7 compares our work with the state-of-art.

Chapter 8 describes how to estimate the relative best-case response time using the independent timing analysis, but only for bursty traffic. Chapter 9 employs the relative WCRT analysis presented in Chapter 5 to determine a minimum bandwidth reservation for Credit-Based Shapers in a single switch. Finally, in Chapter 10, we present conclusions and suggestions for future research.

Chapter 2

Ethernet TSN and Existing Response Time Analyses

This chapter will begin with an introduction to the Ethernet TSN task group. We present a list of completed or in-progress standards to provide an overview of what the Ethernet TSN working group hopes to accomplish and how the evolution process is going.

This dissertation does not address all aspects of Ethernet TSN standards. Rather, it focuses specifically on Credit-Based Shaping, a shaping mechanism that operates at the output port of bridges to regulate the latency and throughput of large-volume network traffic, such as audio and video data (see Section 2.1.2), and the bandwidth reservation for Credit-Based Shaping in the traffic management part of the standards (see Section 2.1.4). However, for completeness, we also provide a brief introduction to other parts of the standard relevant to understanding the general purpose of Ethernet TSN. This introduction helps to comprehend where Credit-Based Shaping fits within the broader framework of Ethernet TSN.

Finally, the overview includes a survey of the existing response time analyses of traffic shaping that aim to mathematically compute bounds on the latency. These papers form the starting point on which this dissertation sets out to improve.

Note that both the overview of Ethernet TSN and existing response time analyses were finalized around July, 2018, prior to the publication of our journal paper Cao et al. (2018b).

2.1 Ethernet TSN

The Ethernet Time Sensitive Networking (TSN) task group develops a set of standards to enable deterministic real-time communication over Ethernet. It was formerly known as the Audio Video Bridging task group, formed in 2007 to deal with the streaming of audio and video data over Ethernet. The Ethernet AVB standards were first applied in professional audio and video studios, and its success attracted the attention of the automotive industry and at a later moment the industrial automation domain. The group has been transformed into the TSN task group since 2012, and the legacy standards established by Ethernet AVB remain, refer to Table 2.1.

The Ethernet TSN task group provides a deterministic service based on Ethernet, a primarily best-effort network. For deterministic service, the network reserves bandwidth and buffering resources according to the agreement between the network and the applications. Thus the network ensures bounded latency, zero congestion loss, and ultra-reliable delivery for these critical traffic flows. It differs from a best-effort service, which works reasonably well for some applications as it delivers the majority of packets most of the time. However, its statistical performance does not guarantee delivery and end-to-end latency and, thus, cannot support industrial and automotive control. The reason to use Ethernet is that the massive increase in the demand for best-effort networking equipment makes it cheaper than other special-purpose digital connections invented in industries. Furthermore, Ethernet's high bandwidth and widespread commercialization make it an ideal candidate for supporting deterministic service.

The use cases for deterministic networking vary substantially. The application includes professional audio and video studios, electrical power generation and distribution, building automation, industrial machine-to-machine, automotive and other vehicle applications, industrial wireless, and cellular radio. The special-purpose digital connections or other alternative networks based on Ethernet (such as EtherCAT, TTEthernet, and Profinet) provide only isolated networks for the exclusive use of a limited number of critical applications. Instead, Ethernet TSN offers flexibility to support dynamic applications through standardized network traffic management.

The Ethernet TSN task group sets up a set of standards which include time synchronization, latency guarantees, reliability, and traffic management to provide deterministic network service. These four parts of the standards are briefly introduced as follows. Refer to Table 2.1 for a list of completed standards and standards under development whose names start with the letter 'P'. A proper selection of standards is required to meet the network requirement of applications.

Table 2.1: Development from Ethernet AVB to Ethernet TSN (ref. to standards dated July 2018)

	Ethernet AVB	Ethernet TSN
Time Synchronization	IEEE 802.1AS: Time Synchronization	IEEE 802.1AS: Time Synchronization P802.1AS-Rev: Time Synchronization
Latency	IEEE 802.1Qav: Credit-Based Shaper	IEEE 802.1Qav: Credit-Based Shaper IEEE 802.1Qbu: Frame Preemption IEEE 802.1Qbv: Scheduled Traffic - Time Aware Shaper IEEE 802.1Qch: Cyclic Queuing and Forwarding P802.1Qcr: Asynchronous Traffic Shaper
Reliability		IEEE 802.1Qca: Path Control and Reservation IEEE 802.1Qci: Per-Stream Filtering and Policing IEEE 802.1CB: Frame Replication and Elimination P802.1AS-Rev: Time Synchronization
Management	IEEE 802.1Qat: Stream Reservation Protocol	IEEE 802.1Qat: Stream Reservation Protocol P802.1Qcc: TSN Configuration P802.1Qcp: YANG Data Model P802.1CS: Link-local Reservation Protocol

2.1.1 Time Synchronization

Time synchronization of all network devices and hosts is essential to achieve deterministic communication with bounded end-to-end latency for time sensitive applications. A robust mechanism to provide global time is the foundation for scheduling time-sensitive traffic through the network components. In Ethernet TSN, this part of the standard is accomplished in close collaboration with the IEEE 1588 working group, which was formed around 2000.

IEEE Std 802.1AS-2011 (Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks) provides a Layer 2 time synchronizing service that is appropriate for the most stringent requirements of consumer electronics applications. It utilizes some variants of IEEE 1588 Precision Time Protocol and eventually becomes a profile of IEEE.

The ongoing project P802.1AS-Rev improves IEEE Std 802.1AS-2011, which is more compatible with IEEE 1588 V3. In P802.1AS-Rev, implementation of synchronization is allowed on any device, not just bridges, but also routers and firewalls. Furthermore, the simultaneous synchronization of two independent time scales is enabled. A universal time is used to time stamp events, production data, or sampled values, whereas a working clock is utilized to synchronize actuators, sensors, or control units. Finally, fault tolerance has been improved by using multiple grandmaster clocks and the multiple connections to these grandmaster clocks in case that a grandmaster becomes defective. Hence, P802.1AS-Rev partly contributes to the improvement of network reliability.

2.1.2 Bounded Latency and Zero Congestion Loss

One of the essential features that a deterministic network provides is bounded end-to-end latency. The end-to-end latency consists of (1) link delay, the transmission time on the physical link between two network components; (2) processing delay, the delay from the reception of the frame at the input port to that the frame has entered into a queue at the output port, which includes the input queuing delay and the fabric switching; (3) output queuing delay: the time spent from the insertion of the frame into the queue to that the frame finishes its transmission for output on the link. If preemption (IEEE 802.3Qbu) is enabled, the additional interrupted time should also be considered. As the output queuing delay contributes to the most variations of the end-to-end latency, this part of the standards mainly improves the queuing algorithms at the output port.

The queuing algorithms are mostly defined in IEEE Std 802.1Q. Strict priority and weighted fair queuing are commonly used but can not guarantee bounded latency. Ethernet TSN proposes a hierarchical queuing algorithm;

i.e., the network traffic is classified according to its priority, and the traffic flow of each priority is individually shaped. The suitability of the specific shaping mechanism for a deterministic network depends on whether it is possible to compute the worst-case latency and thus allocate sufficient buffering resources to assure zero congestion loss. Following is a brief overview of shaping mechanisms that have been standardized or are in the standardization process.

Credit-Based Shaping (CBS) is defined by IEEE 802.1Qav-2009 and now clause 35 of IEEE Std 802.1Q-2014 by the former Ethernet AVB group. It is the first standardized shaping mechanism that regulates large-volume network traffic, such as audio and video data, through credit variation according to the reserved bandwidth.

Time-Aware Shaping (TAS) is defined by IEEE Std 802.1Qbv-2015 with the aim to support time-sensitive traffic by introducing time-triggered transmission on top of the other traffic classes. This time-triggered transmission focuses on small amounts of information, such as control traffic, that need to be communicated with very low and predictable latencies to support industrial automation, process control, and vehicle control.

The Time Aware Shaper defines two transmission gate states, 'open' and 'closed', and a network schedule defines the gate operation. The time-critical messages are selected for transmission when their corresponding gate is open and the rest are closed. Guard bands or preemption mechanisms can be applied to strictly prevent critical traffic from getting blocked by other network traffic. TAS functions like TDMA and provides the most stringent shaping mechanism to provide deterministic latency for time-critical applications.

Transmission Preemption is defined in IEEE Std 802.1Qbu and IEEE Std 802.3br (Interspersing Express Traffic Task Force). In these standards, the queues at the output port are designated as 'preemptable' and 'preemptive'. Frames from preemptable queues can be stopped for transmission if a preemptive queue is selected for transmission and resume transmission when the preemptive queue finishes its transmission. It reduces the interference a preemptable queue imposes on a preemptive queue from a maximum frame size to a maximum fragment size. This way, the preemption allows immediate transmission and ensures the minimum latency of time-critical traffic, and the non-critical traffic is fragmented to use the remaining bandwidth.

Cyclic Queuing and Forwarding (CQF) is defined in IEEE Std 802.1 Qch-2017 together with IEEE Std 802.1 Qci-2016 (Per-stream Filtering and Policing). CQF, formerly known as Peristaltic Shaping, is a shaping mechanism that delivers deterministic and easily calculated latency for time-sensitive traffic streams. The stream traffic is transmitted and queued for transmission along a network path in a cyclic manner. The latency for a frame that transits the network is expected to be described by the cycle time and the number of hops. This is unaffected by the topology and the interference from other non-time-sensitive traffic.

The input port accepts the frame based on an input schedule, which is set to assign frames in synchronization with the output schedule at the output port. The output port involves the use of two transmission queues. During even-numbered cycles, queue 1 accumulates received frames from the input ports (and does not transmit them), while queue 2 transmits any queued frames from the previous odd-numbered cycle (and does not receive any frames). During odd-numbered cycles, queue 2 accumulates received frames from the input ports, while queue 1 transmits any queued frames from the previous even-numbered cycle.

The latency of CQF depends on the clock synchronization between the input and output schedule and the selection of the cycle time. A small cycle time gives better latency and latency variation but limits the traffic flow; a big cycle time enhance the traffic flow but also increases the latency and latency variation.

Asynchronous Traffic Shaping (ATS) is under development in the new project, P802.1Qcr. This project specifies procedures and managed objects for a bridge to perform asynchronous traffic shaping over full-duplex links with constant bit data rates. This work intends to develop a new shaping mechanism that does not rely on synchronous communication, thereby providing independence from clock synchronization mechanisms and higher link utilization than synchronous mechanisms while meeting the requirement of bounded latency and zero congestion loss.

2.1.3 Reliability

Another essential feature deterministic networking provide is ultra-reliable delivery. As mentioned in the previous subsection, zero congestion loss is ensured by allocating sufficient buffering resources. The other important cause of frame loss is equipment failure. The strategy the Ethernet TSN task group adopts is sending multiple copies of a sequence-numbered data stream over multiple paths, eliminating duplicates at or near the destina-

tions. There is no failure detection and recovery mechanism. In this way, a single random event or a single equipment failure does not introduce frame loss. However, if equipment failures occur on each redundant path, there will be frame loss, and this loss can not be recovered if the equipment failures remain unfixed.

IEEE Std 802.1Qca-2016 defines how to set up explicit forwarding paths and integrate a bandwidth and stream reservations tool along the forwarding paths. IEEE Std 802.1CB describes how to replicate frames with a sequence number added, as well as how to identify and eliminate duplicate frames at the relay nodes or end stations.

IEEE Std 802.1Qci, as mentioned in CQF shaping, performs frame counting, filtering, policing, and service class selection at the input ports. It also includes detecting and mitigating disruptive transmissions in a network, thus improving the network's reliability. The ongoing project P802.1AS-Rev improves the reliability of clock synchronization in the network.

2.1.4 Traffic Management

Flexibility is the most radical change that Ethernet TSN aims to achieve compared to existing network communication protocols for supporting real-time applications. The task group designs a set of tools for network configurations to support and monitor hierarchical traffic flow. In this way, proper functioning of critical flows can be maintained at all times in a dynamic network environment, i.e., new agreements between the network and applications can be added, and the old ones revoked.

IEEE Std 802.1Qat-2010 (clause 34 of 802.1Q-2014) defines the Stream Reservation Protocol (SRP) to support forwarding and queuing requirements for time-sensitive streams. It provides a means to detect the boundary between a set of bridges that support SRP and surrounding bridges that do not support SRP. This bandwidth reservation mechanism only supports streams that undergo Credit-Based Shaping.

IEEE Std 802.1Qcc-2018 provides a mechanism to improve the existing SRP given in IEEE Std 802.1Qat in order to meet the requirements of industrial and automotive systems. These include support for more streams, more configurable SR (Stream Reservation) classes, a better description of stream characteristics, support for Layer 3 streaming, deterministic stream reservation convergence, and UNI (User Network Interface) for routing and reservations. These network configurations can be achieved statically or dynamically.

The ongoing project P802.1 Qcp adopts YANG (Yet Another Next Gen-

eration), a formalized data modeling language that can be used by NETCONF, to provide hierarchical configurable and operational data structures. As YANG is a widely accepted protocol that is used to simplify network configuration, the development of the YANG data model provides for manageable entities specified in IEEE Std 802.1Q. It will give general support to the industry.

Another ongoing project P802.1CB creates a Link-local Registration Protocol (LRP) to replicate a registration database from one end to the other of a point-to-point link and to replicate changes to parts of that database. It aims to facilitate the creation of application protocols that distribute information through all or part of a network.

2.2 Existing Works on Response Time Analysis

In this section, we first discuss previous research works that focus on the timing analysis of Credit-Based Shaping in Ethernet TSN. We focus particularly on their methodology, their coverage of single- or multi-hop networks, their assumptions on interfering traffic, and the tightness of their approach. Following that, we present a brief overview of other works relating to traffic shaping techniques in the broader scope of Ethernet TSN. Note that we only consider works that were published before our journal paper Cao et al. (2018b).

Considering only Credit-Based Shaping in Ethernet TSN, various schedulability analysis techniques have been proposed to compute the worst-case response times for classes A and B in Ethernet TSN, such as Imtiaz et al. (2009) using delay computation for a single hop, Lee et al. (2006) and De Azua and Boyer (2014) using the network calculus of Leboudec and Thiran (2001). However, these analysis techniques aim at computing the worst-case per class, instead of per individual message. As it is vital for many application domains to know the worst-case delay of individual messages in a TSN network, other analysis techniques have been proposed to bound the delay of each message in TSN networks. For instance, the analysis presented in Reimann et al. (2013) calculates the message delay using Modular Performance Analysis (MPA) of Wandeler et al. (2006).

Moreover, Diemer et al. (2012a,b) present an extensive formal analysis of Ethernet AVB based on busy period analysis (Davis et al. (2007); Bril et al. (2009); Lehoczky (1990)). It uses the Compositional Performance Analysis approach (Henia et al. (2005)) to take an output stream of one component and turn it into an input stream of a connected component for multi-hop network analysis. However, as the busy period analysis is initially designed for non-idling servers, the application toward Credit-Based Shaping still

faces certain challenges. For instance, pessimism remains a concern in these works. Axer et al. (2014) and Bordoloi et al. (2014) independently improve on the work of Diemer et al. (2012b) to reduce the existing pessimism. Bordoloi et al. (2014) attempts to address two sources of pessimism: one source is that the high-priority traffic is also shaped, and the second is that the high-priority traffic which transmits during idling time should not be counted as interference. This improvement is limited to a single-hop setting, while Axer et al. (2014) attempts to address part of that same improvement but considers a multi-hop analysis.

Ashjaei et al. (2017) presents a worst-case response time analysis, which extends the work of Bordoloi et al. (2014) with support for scheduled traffic and further proposes a methodology for allocating network bandwidth to Credit-Based Shapers. The improvement made by Bordoloi et al. (2014) is not considered in Ashjaei et al. (2017), as the improved solution is too complicated to be applied to determine the network bandwidth reservation. Li and George (2017) presents a schedulability analysis to compute the worst-case delay of messages from classes A and B using the trajectory approach (Martin and Minet (2006)) in a multi-hop network. This work partially uses busy period analysis to limit the search space of time within which the worst-case delay upper bound can be found, and it provides a tighter analysis compared to Diemer et al. (2012b).

Besides the pessimism, another disadvantage of using busy period analysis is that a detailed traffic model is required, e.g., a periodic/sporadic model in Bordoloi et al. (2014) and Li and George (2017) or arrival curves in Diemer et al. (2012b) and Axer et al. (2014). The dependence on a detailed traffic model makes the timing analysis less robust against traffic flow changes.

In the scope of TSN networks considering other shapers than the Credit-Based Shaper, several works have addressed the schedulability analysis of traffic classes A and B. For instance, an analysis is given in Maxim and Song (2017) to compute the worst-case delay of messages in TSN considering the time-aware shaper, which is based on the technique proposed in Cao et al. (2016a). It should be noted that both Maxim and Song (2017) and Cao et al. (2016a) considered a single-switch architecture. Moreover, the work presented in Thiele et al. (2015) proposed an analysis technique for time-aware shaper and peristaltic shaper in TSN networks.

Since 2012, the work on Credit-Based Shaping has become part of a larger standardization initiative in the Ethernet TSN task group. In Ethernet TSN, also other shaping technologies are being developed (see e.g. Thangamuthu et al. (2015)). Several works have presented timing analy-

ses for these new shapers. For instance, Thiele et al. (2015) has proposed an analysis technique based on busy period analysis for the time-aware shaper and peristaltic shaper, and Thiele and Ernst (2016) has addressed the burst-limiting shaper also using busy period analysis. There are also studies addressing the schedulability analysis of Credit-Based Shaping with support for scheduled traffic for a single-switch architecture, e.g. Maxim and Song (2017) and Ashjaei et al. (2017).

In this dissertation, we only focus on Credit-Based Shaping and do not cover the new shapers and the combination of shapers in Ethernet TSN. However, it should be noted that the work of Maxim and Song (2017) is based on the relative worst-case response time analysis techniques developed in chapter 5 of this dissertation, which first appeared in Cao et al. (2016b) and Cao et al. (2016a). Furthermore, after the main body of work presented in this dissertation was finished, some other works that make use of our approach have appeared. A few of those will be briefly discussed in the future work section of Chapter 10.

Chapter 3

System Model

In this section, we first sketch the mechanics of Credit-Based Shaping, relevant for our analysis. Subsequently, we introduce some notation to capture these mechanics. Finally, we give mathematical relationships that characterize the mechanics of Credit-Based Shaping formally. For this, we adopt an axiomatic approach. This means that, instead of describing exactly how a shaper behaves by presenting a construction of its executions, we only specify relations between events and variables that an implemented Ethernet TSN switch should respect. Subsequently, we use these relations to derive properties of the behavior of a shaper. Naturally, in some places we provide explicit constructions of executions, either as illustrations, or as a witness that a derived bound can be achieved in tightness proofs. As a side-effect, these constructions also serve to convince the reader that the axioms we posed are not self-contradictory.

3.1 Mechanics of Credit-Based Shaping

We are modeling the behavior of a single switch that is part of an Ethernet TSN network. The goal of this switch is to pass on frames from its inputs to its outputs according to some routing table. However, we are interested only in the time it takes to perform this task, and not in the actual routing of frames. In order to be able to calculate this, it is important to know how a switch schedules its transmissions.

As a basis, Ethernet TSN adopts non-preemptive strict priority scheduling. Frames are gathered in one queue for each priority class, and a frame can start its transmission only if there is no high-priority queue that can start (strict priority), if there are no earlier frames of the same priority

in the queue (FIFO), and if there is currently no transmission going on (non-preemptive), see Figure 3.1.

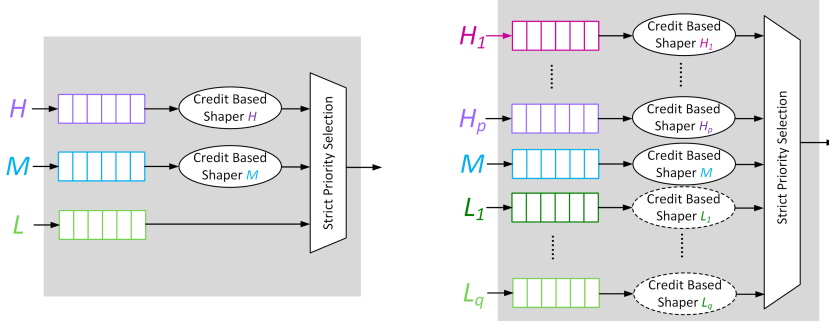


Figure 3.1: The output port of an Ethernet TSN switch. Left panel: priority classes H and M undergo Credit-Based Shaping, while class L obtains the remaining bandwidth. Right panel: priority classes H_1, H_2, \dots, H_p and M undergo Credit-Based Shaping. Whether classes L_1, L_2, \dots, L_q are under Credit-Based Shaping (shown as dotted lines) is irrelevant in the WCRT analysis of class M , which will be further explained.

On top of this, Ethernet TSN adds a Credit-Based Shaping mechanism for the priority class to limit traffic bursts. In order to start the transmission of a frame, the priority class of that frame can not have negative *credit*. Whenever credit is above or equal to 0, a frame can start transmission, assuming the other rules also allow this. If credit is lower than 0, frames from low-priority classes are allowed to transmit. Credit for a priority class X starts at 0, and while a frame is in transmission it drops at a rate of α_X^- . If no frame is being transmitted and credit is negative, no further frames of that class can be transmitted and credit rises with a rate of α_X^+ . Also, if class X has pending frames waiting but cannot transmit, credit rises at a rate of α_X^+ . Positive credit is reset to 0 if the queue of class X does not contain any frames anymore. One example of Credit-Based Shaping is illustrated in Figure 3.2.

In the standard, α_X^+ is the bandwidth reservation of class X , with $0 < \alpha_X^+ \leq BW$, where BW is the total bandwidth. α_X^- is the remaining bandwidth: $\alpha_X^- = BW - \alpha_X^+$. We observe that the maximum utilisation for class X is given by $\frac{\alpha_X^+}{BW}$, see also Figure 3.2. When multiple streams are of the same priority class, they share the bandwidth of that class.

When designing a system it is reasonable to expect that the total reservation does not exceed the total available bandwidth, resulting in the re-

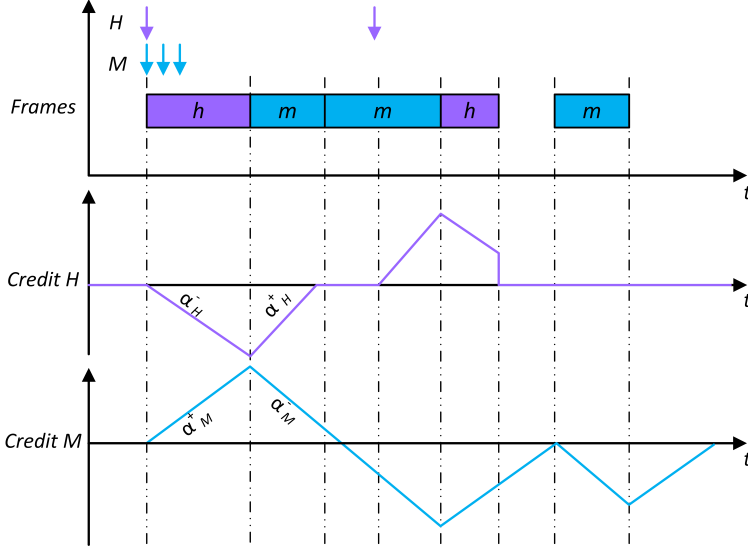


Figure 3.2: An example of Credit-Based Shaping with class H and class M traffic. Arrows here indicate frame arrivals while rectangles indicate the transmission of frames. The transmission follows two rules: H is prioritized over M , and there must be non-negative credit at the start of a transmission. The rising slope of credit α^+ is set to the reserved bandwidth and the falling slope α^- is set to the remaining bandwidth. For instance, if α_H^+ is set to 60 Mbps while the total available bandwidth is 100 Mbps, then α_H^- is set to 40 Mbps, which leads to a maximum utilization of H of 60%. Note that the horizontal axis is at credit 0.

quirement:

$$\sum_{X \in \mathbb{P}} \alpha_X^+ \leq BW, \quad (3.1)$$

where \mathbb{P} denotes the set of all priority classes. Although this requirement is strictly speaking not enforced by the standard, it seems a reasonable design choice in general, and we do adopt this assumption later in our analysis. The only exception to this rule, is the lowest priority class, which in practice is not shaped at all, but can be treated as if it is shaped with a reservation equal to the bandwidth ($\alpha_X^+ = BW$).

Another parameter set in the standard, is the maximum size C_X^{\max} that a frame of a given priority class X may have. Size, in this case, refers to the number of bits that need to be transmitted, but for ease of presentation is converted to the maximum amount of time a transmission takes in this dissertation.

3.2 Notation

Priority. As indicated above, we use the capital letter X , but also Y , M , H , and L , to denote priority classes of streams, and we write \mathbb{P} for the set of all priority classes. This set is considered to be totally ordered, so we may write $X, Y \in \mathbb{P}$ and $X < Y$ to indicate that X has a lower priority than Y . Furthermore, in the remainder, we sometimes write $\mathbb{X} \subseteq \mathbb{P}$ if we want to consider a subset of priority classes \mathbb{X} , and write $\alpha_{\mathbb{X}}^+ = \sum_{X \in \mathbb{X}} \alpha_X^+$ and $\alpha_{\mathbb{X}}^- = BW - \alpha_{\mathbb{X}}^+$ to indicate the cumulative reservation of such a set and its remainder.

Frames, streams and sources. To indicate that a frame x is of priority class X we write $x \in X$. The set of all frames is denoted by \mathbb{F} , so that formally we can regard a priority class $X \subseteq \mathbb{F}$ as a set of frames, as well as an element of \mathbb{P} . However, when discussing the behavior of Credit-Based Shaping, it is more natural to speak of a *stream* X when considering the flow of frames from input to transmission, and to speak of a *priority class* X when considering scheduling decisions on queues. We will use the word *source*, and write $\tau \subseteq X$, to indicate a specific set of frames of the same priority class under investigation, e.g., a set of periodically-generated frames.

Dynamics and credit. As mentioned before, we follow an axiomatic approach to formally model the dynamics of Credit-Based Shaping. In this approach, we consider executions u , and give properties under which u is considered a valid execution of the system, denoted $u \in \mathbb{U}$. These properties are in terms of events and variables defined by the standard. For example, we write $CR_X^u(t)$ to denote the credit of priority class X at time $t \in \mathbb{R}_0^+$ during an execution u . As an example of a property, we have that $CR_X^u(0) = 0$ for all $u \in \mathbb{U}$, saying that at the start of execution (time 0) the credit of every shaper is 0.

More on executions. Typically, one may consider an execution u to be an ordered set of observations, like events or valuations of variables. But for most of our analysis, it is not important how executions are exactly defined, as long as we can define the necessary functions that keep track of variables and events in the system. The standard does not prescribe the exact implementation of the traffic shapers, it merely gives a specification of possible implementations. Therefore, we strive not to define here what an execution is exactly. It is sufficient to know which axioms need to hold for an execution. This keeps our analysis independent of the chosen implementation. Sometimes we do give examples of valid executions by specifying

which events and variable valuations are observed in which order. The purpose of these is either to serve as illustrations, or as witnesses to prove that an execution with certain properties actually exists.

Arrival time. The behavior of a switch is determined by the arrival time of frames during an execution, and by the order of arrival in case multiple frames arrive at the same point in time. We assume that for any given set of frames we are able to distinguish the first frame that arrived. Technically, let $\mathbb{F}^u \subseteq \mathbb{F}$ denote those frames that arrive during an execution u , and assume (reasonably) that \mathbb{F}^u is a countable set. We denote the arrival time of an arbitrary frame $x \in \mathbb{F}^u$ by $a^u(x) \in \mathbb{R}_0^+$. Furthermore, because it may be the case that multiple frames arrive at the same point in time, we sometimes write $x <_u x'$ to emphasize the order in which frames arrive. In particular, for some $x, x' \in \mathbb{F}^u$ we may use $x <_u x' \wedge a^u(x) = a^u(x')$ to denote that x arrived before x' at the input queue, but their arrival times cannot be distinguished.

Start, finish, and worst-case response time. For a given frame $x \in \tau$, the response time in an execution u is defined as the difference between finish time $f^u(x)$ of that frame and its arrival time $a^u(x)$. The worst-case response time of a frame x given a set of possible executions \mathbb{U} is defined as:

$$WR^{\mathbb{U}}(x) = \sup_{u \in \mathbb{U}} f^u(x) - a^u(x). \quad (3.2)$$

The worst-case response time of a source τ can be further defined as:

$$WR^{\mathbb{U}}(\tau) = \sup_{x \in \tau} WR^{\mathbb{U}}(x). \quad (3.3)$$

In this dissertation, we present a relative response time analysis, meaning that we compare an execution u and its corresponding uninterfered execution u_0 without inter-priority traffic, which is considered interference, to calculate the worst-case relative delay.

Definition 3.1 (Uninterfered execution). *Given a priority class X and an execution u , we define the uninterfered execution of X to be an execution u_0 (dependent on u) in which only frames of X arrive, and arrive at the same time as in u . Furthermore, they have the same transmission time. So $\mathbb{F}^{u_0} = \mathbb{F}^u \cap X$ and for all $x \in \mathbb{F}^{u_0}$ we have $a^{u_0}(x) = a^u(x)$ and $C^{u_0}(x) = C^u(x)$. Assuming the scheduling of frames is deterministic, this u_0 is uniquely defined.*

Given non-preemptive transmissions in the switch, the finish time $f^u(x)$ is determined by the start time $s^u(x)$, hence we have $f^u(x) = s^u(x) + C^u(x)$.

We can then derive a bound on the response time of a frame x in any execution u from that of u_0 by studying the delay in start times as shown below:

$$\begin{aligned}
& WR^{\mathbb{U}}(x) \\
&= \sup_{u \in \mathbb{U}} f^u(x) - a^u(x) \\
&= \sup_{u \in \mathbb{U}} s^u(x) + C^u(x) - a^u(x) \\
&= \sup_{u \in \mathbb{U}} s^u(x) + C^{u_0}(x) - a^{u_0}(x) \\
&= \sup_{u \in \mathbb{U}} s^{u_0}(x) + C^{u_0}(x) - a^{u_0}(x) + (s^u(x) - s^{u_0}(x)) \\
&\leq \left(\sup_{u \in \mathbb{U}} s^{u_0}(x) + C^{u_0}(x) - a^{u_0}(x) \right) + \left(\sup_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x)) \right) \\
&= WR^{\mathbb{U}_0}(x) + \left(\sup_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x)) \right),
\end{aligned} \tag{3.4}$$

where $WR^{\mathbb{U}_0}(x)$ denotes the worst-case response time of x in the set $\mathbb{U}_0 = \{u_0 \mid u \in \mathbb{U}\}$ of all uninterfered executions, while the term $\sup_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x))$ is called the *worst-case relative delay* of a frame x . In Chapter 5, we seek an upper bound on this relative delay, given that $x \in \tau \subseteq X$ is a frame from a given source or priority class. It is assumed that the worst-case response time of a frame x in the uninterfered execution $WR^{\mathbb{U}_0}(x)$ is known to determine the absolute worst-case response time $WR^{\mathbb{U}}(x)$ with this relative analysis.

It's worth noting that we analyze the relative delay without restricting a^u to any particular arrival pattern. In particular, we do not assume the usual periodic or sporadic arrival. Only in Chapter 7, when comparing our results to other works that do rely on periodic/sporadic arrivals, do we assume periodic behavior for a_0^u to calculate $WR^{\mathbb{U}_0}(x)$.

Now, before proceeding to the relative analysis, we develop the relations required to derive this upper bound by further formalizing the mechanics of Credit-Based Shaping.

3.3 Formalization of Shaping Rules

In this section, we rephrase the earlier descriptions of Credit-Based Shaping in terms of mathematical relationships between arrival times, order of arrival, credit, transmission times, start times, and finish times. We use these formalized relationships in the later chapters to bound the worst-case response time of frames under Credit-Based Shaping. As mentioned before,

we adopt an axiomatic approach to define the behavior of Credit-Based Shaping. The predicates in this section define which executions u are valid elements of the set of all executions \mathbb{U} of a Credit-Based Shaper. Next, we can use these predicates to prove properties of these executions in the analysis.

Non-preemptive. Let us start by recalling a property mentioned previously. The fact that transmissions are non-preemptive means that for all $x \in \mathbb{F}$ and executions $u \in \mathbb{U}$ we find:

$$f^u(x) = s^u(x) + C^u(x). \quad (3.5)$$

Furthermore, as we have seen before, the fact that the standard bounds the maximum size of frames $x \in X$ gives us a bound on the transmission time:

$$C^u(x) \leq C_X^{\max}. \quad (3.6)$$

Single Transmission. Since we assume a single transmission at a time, two frames cannot start their transmission at the same time and any frame that starts earlier should finish its transmission before the next frame starts. For all $x, x' \in \mathbb{F}$ and $u \in \mathbb{U}$:

$$x \neq x' \Rightarrow s^u(x) \neq s^u(x'), \quad (3.7)$$

and

$$s^u(x) < s^u(x') \Rightarrow f^u(x) \leq s^u(x'). \quad (3.8)$$

Non-negative credit. Next, we study the start time of frames in more detail. In particular, we have the rule that in order to transmit a frame, its priority class must not have negative credit. So for any priority class $X \in \mathbb{P}$ and frame $x \in X$ we find in any execution $u \in \mathbb{U}$:

$$CR_X^u(s^u(x)) \geq 0. \quad (3.9)$$

FIFO. Furthermore, the fact that frames of the same class are queued and transmitted in the order in which they arrive (First-In First-Out), gives us for all executions $u \in \mathbb{U}$, all $X \in \mathbb{P}$, and all $x_1, x_2 \in X$:

$$x_1 <_u x_2 \Rightarrow s^u(x_1) < s^u(x_2). \quad (3.10)$$

Strict priority. Similarly, the general notion that frames of higher priority take precedence if credit allows can be formalized by stating that for all $u \in \mathbb{U}$, all $X, Y \in \mathbb{P}$ with $X < Y$, and all $x \in X$ and $y \in Y$ we have

$$(a^u(y) \leq s^u(x) \wedge CR_Y^u(s^u(x)) \geq 0) \Rightarrow s^u(y) \leq s^u(x). \quad (3.11)$$

Note that Eq.(3.11) captures the notion of priority even in the case of multiple simultaneous transmissions. Combining this with Eq.(3.7) and Eq.(3.8) we get:

$$(a^u(y) \leq s^u(x) \wedge CR_Y^u(s^u(x)) \geq 0) \Rightarrow f^u(y) \leq s^u(x). \quad (3.12)$$

From Eq.(3.12) it follows that, if a higher priority has pending load and credit, a lower priority cannot start its transmission. If it would start, the predicate would be violated.

Eager scheduling. If we combine all of the previous properties (Non-preemptive, Single transmission, Non-negative credit, FIFO and Strict priority) with the assumption that frames are scheduled as soon as credit allows, we obtain the infimum over the above properties as the start time of a frame. For all $u \in \mathbb{U}$, $X \in \mathbb{P}$ and $x \in X$ we have:

$$s^u(x) = \inf \left\{ t \left| \begin{array}{l} \forall_{z \in \mathbb{F}} s^u(z) < t \Rightarrow f^u(z) \leq t \\ CR_X^u(t) \geq 0 \\ \forall_{x' \in X} x' <_u x \Rightarrow s^u(x') < t \\ \forall_{y \in Y > X} (CR_Y^u(t) \geq 0 \wedge a^u(y) \leq t) \Rightarrow (f^u(y) \leq t) \end{array} \right. \right\}. \quad (3.13)$$

Credit at start. The only unknown in the equation above, is the credit, for which we already mentioned that it starts at 0 in every execution $u \in \mathbb{U}$ and for any priority class $X \in \mathbb{P}$:

$$CR_X^u(0) = 0. \quad (3.14)$$

Credit during transmission. While a frame of priority class $X \in \mathbb{P}$ is in transmission, credit drops at a rate of α_X^- . Therefore, for any two points in time $t \leq t'$ for which there exists a frame $m \in X$ such that $s^u(m) \leq t$ and $t' \leq f^u(m)$ we find:

$$CR_X^u(t') = CR_X^u(t) - \alpha_X^- \cdot (t' - t). \quad (3.15)$$

Credit recovery. While credit of priority class $X \in \mathbb{P}$ is negative and X is not transmitting, credit rises with a rate of α_X^+ . Therefore, for any two points $t \leq t'$ with $t' \leq t - \frac{CR_X^u(t)}{\alpha_X^+}$ and X not transmitting at t (i.e., $\forall_{x \in X} f^u(x) < t \vee s^u(x) > t$) we find:

$$CR_X^u(t') = CR_X^u(t) + \alpha_X^+ \cdot (t' - t). \quad (3.16)$$

Pending load. Similarly, if there are frames in the queue of priority class $X \in \mathbb{P}$, but this class cannot transmit, this also leads to a credit rise with a rate of α_X^+ . We say that a priority class X has *pending load* at time $t \in \mathbb{R}_0^+$, denoted $Pending_X(t)$, if there exists a frame $m \in X$ such that $a^u(m) \leq t$ and $f^u(m) > t$. Furthermore, we say that a priority class Y is *transmitting* at time $t \in \mathbb{R}_0^+$ if there exists a frame $n \in Y$ such that $s^u(n) \leq t$ and $f^u(n) > t$. Now, for any two points $t \leq t''$ such that X has pending load at t but some other class Y is transmitting at every point t' with $t \leq t' < t''$, we find:

$$CR_X^u(t'') = CR_X^u(t) + \alpha_X^+ \cdot (t'' - t). \quad (3.17)$$

Credit reset. Finally, we know that credit of priority class $X \in \mathbb{P}$ is reset as soon as there is no pending load. Therefore, at any time $t \in \mathbb{R}_0^+$ we find:

$$\left(\neg Pending_X(t) \wedge \lim_{t' \uparrow t} CR_X^u(t') \geq 0 \right) \Rightarrow CR_X^u(t) = 0. \quad (3.18)$$

Definition 3.2 (Credit-Based Shaping). *An execution $u \in \mathbb{U}$ is Credit-Based Shaped if its arrival, start, and finishing times a^u, s^u and f^u satisfy Eq. (3.5) to Eq. (3.18).*

The relations above formally and thoroughly characterize the behavior of Credit-Based Shaping. During the analysis, we discovered that it is more convenient to introduce and use the definition of *Eligible Intervals* for this study. The concept of *Eligible Intervals* is tailored to idling servers, which simplifies the relative worst-case response time analysis. Detailed presentation of this concept will be provided in the next chapter.

Chapter 4

Eligible Intervals

The notion of *Eligible Interval* was firstly introduced in Cao et al. (2016b), where we studied the Credit-Based Shaping behavior for two types of interfering traffic: one low priority or one shaped high priority. The result showed that tight worst-case response time bounds were more easily obtained compared to the conventional busy-period analysis (Bordoloi et al. (2014); Diemer et al. (2012b)) given the assumed interfering traffic.

We attribute the tightness as well as the simplicity of our analysis to the use of Eligible Intervals, during which there is both pending load and credit available to a shaper. The use of Eligible Intervals takes the idling nature of Credit-Based Shaping into account, which further leads to a separation of concerns in which we firstly relate the start of Eligible Intervals to a known uninterfered schedule, and secondly relate the credit at the start of a transmission to the relative delay caused by the interfering traffic. Subsequently, we can easily produce a bound on the worst-case relative delay by further studying the maximum attainable credit. This approach remains valid when we address the more complex case in which the interference may be a combination of both high- and low-priority traffic (one high priority is addressed in Cao et al. (2016a) and multiple high priorities is addressed in Cao et al. (2018b)).

In this chapter, we firstly introduce the general definition of Eligible Interval for idling servers and the instantiation of this definition exclusively for Credit-Based Shaping in Section 4.1. Secondly, in Section 4.2, we assume a single input stream of a single priority class X , i.e., a known uninterfered execution, and examine the effect of Credit-Based Shaping on the output stream by calculating the start times as the result of shaping. This results in an ordinary FIFO schedule, scaled to the reserved bandwidth. Lastly,

we compare an interfered schedule to its uninterfered one to prove the two essential theorems. The first states that *the start time of an Eligible Interval in an interfered schedule is always earlier than or equal to the start time of the Eligible Interval's first frame in an uninterfered schedule*, see Section 4.3, and the second states that *any experienced relative delay is proportionally bounded by the amount of credit at the start of transmission*, see Section 4.4. Since we do not restrict the interference to any pattern, i.e., the interfering traffic may be periodic, sporadic, bursty, or of any other arrival pattern, the two theorems are applicable when we address the interference of multiple high priorities with the presence of low-priority interfering traffic in our analysis. They are potentially extensible to more complex combinations of interference. We discuss the final result, bounding the maximum delay by bounding the maximum attainable credit, in the next chapter.

4.1 Eligible Intervals

We firstly introduce the general definition of Eligible Intervals, suitable for idling servers. Unlike the busy period that only considers the pending load, an Eligible Interval takes both the pending load and transmission rules into account, see the following definition.

Definition 4.1 (Eligible Interval). *Given an execution u , a priority class X is eligible for transmission at a time t if it has a pending load, and it either satisfies the rules of its shaper to start transmission or it is in transmission. An Eligible Interval of X is a maximal interval during which X is eligible.*

As we focus only on Credit-Based Shaping in this dissertation, we tailor Definition 4.1 exclusively for Credit-Based Shaping as in Definition 4.2. We also give specific rules to determine the start and end of an Eligible Interval for Credit-Based Shaping strictly following the definition.

Definition 4.2 (Eligible Interval for CBS). *Given an execution u , a priority class X under Credit-Based Shaping is eligible for transmission at a time t if it has pending load, and it either has non-negative credit, or is in transmission at this time t . An Eligible Interval of X is a maximal interval during which X is eligible.*

For mathematical convenience, we consider transmissions to occur in left-closed and right-open intervals, which means that eligible intervals are also left-open and right-closed. They start with an arrival or with credit recovering to zero, and end at the end of a transmission, having no pending load or negative credit. In other words, an interval $E^u = [E_s^u, E_e^u)$ is an Eligible Interval of $X \in \mathbb{P}$ during execution $u \in \mathbb{U}$ iff:

1. $CR_X^u(E_s^u) = 0$ and $Pending_X^u(E_s^u)$;
2. $CR_X^u(E_s^u - \epsilon) < 0$ or $\neg Pending_X^u(E_s^u - \epsilon)$, for a vanishingly small $\epsilon > 0$;
3. For some frame $x \in X$ we have $f^u(x) = E_e^u$;
4. $CR_X^u(E_e^u) < 0$ or $\neg Pending_X^u(E_e^u)$;
5. E_e^u is the smallest time strictly larger than E_s^u with these properties.

For simplicity, in the remainder of the dissertation, we denote the Eligible Interval as $E = [E_s, E_e]$ without the superscript u . It has to be noted that an Eligible Interval is always determined by a given execution u .

4.2 Shaping an Uninterfered Execution

We start our analysis by considering a single stream X , and at this point assume there is no interference from other streams yet. In other words, for the time being we consider only executions in \mathbb{U}_0 (i.e., uninterfered executions). To emphasize this, we will simply write u_0 for the uninterfered execution under consideration in the remainder of this section. In the following sections, we will consider how to analyze more complex executions in which there is high- and low-priority interference.

The first frame $x_0 \in \mathbb{F}^{u_0}$ in this stream arrives at $a^{u_0}(x_0) \geq 0$ and the credit is initially 0 (Eq. (3.14)). A frame can only start transmission when the credit is non-negative (Eq. (3.9)). Since positive credit is possible only when there is interference (only Eq. (3.16) and Eq. (3.17) in Section 3.3 allow credit to rise), frames in an uninterfered stream start their transmission with a credit of 0. During the transmission, the credit decreases at a rate of α_X^- . The credit has to recover to zero before any next frame can be transmitted. The recovery time is the product of the frame duration and the ratio α_X^-/α_X^+ (see Figure 4.1). This gives us the following recurrence for start times.

Property 4.1. *Given a single shaped stream X in an uninterfered execution u_0 , and considering the frame arrivals as a sequence x_i , then the start times are determined by*

$$\begin{aligned}
 s^{u_0}(x_0) &= a^{u_0}(x_0), \\
 s^{u_0}(x_{i+1}) &= \left(s^{u_0}(x_i) + C^{u_0}(x_i) \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+} \right) \right) \max a^{u_0}(x_{i+1}).
 \end{aligned} \tag{4.1}$$

Property 4.1 shows the shaping effect on a single stream to be the same as that of a basic FIFO schedule with enlarged execution times by a factor $(1 + \alpha_X^-/\alpha_X^+)$.

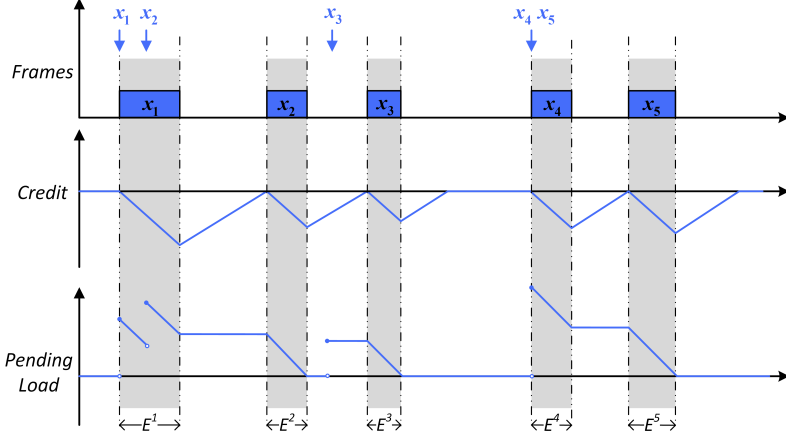


Figure 4.1: The transmission of a single stream X undergoing Credit-Based Shaping. Arrows indicate the frame arrivals, and the transmission under Credit-Based Shaping is shown along with the evolution of the credit and pending load. The amount of pending load of a class is the total remaining transmission time of pending frames of that class. Each Eligible Interval is labeled with E , within which there is only one frame.

For a single uninterfered stream X , the intervals $E^i = [s^{u_0}(x_i), s^{u_0}(x_i) + C^{u_0}(x_i))$ represent a series of eligible intervals of X , illustrated in Figure 4.1. In all our figures that display frame transmission, Eligible Intervals are marked in gray. Note, that when studying an uninterfered execution, we find the shape of Eligible Intervals is rather simple. There is only one X frame in each Eligible Interval, since the credit becomes negative when a frame is sent and no frame can transmit before the credit recovers to zero. Later, we will see that in cases of interference, multiple X frames may transmit in an eligible interval.

Figure 4.1 also shows that if arrival times are late there can be some additional slack between two Eligible Intervals, e.g., between E^3 and E^4 . However, in the uninterfered execution, the minimum distance between $s^{u_0}(x_i)$ and $s^{u_0}(x_{i+1})$ is $C^{u_0}(x_i)(1 + \alpha_X^-/\alpha_X^+)$. This is captured in the following corollary, which can be derived by a proof of induction using Property 4.1 as the induction step. We use this property later in the proof of Theorem 4.1 to determine the start time of an Eligible Interval in an interfered execution.

Corollary 4.1. *Given a single shaped stream X in an uninterfered execution u_0 , and considering the arrivals of frames as a sequence, we find for*

all $x_0 \dots x_{i+k} \in X$:

$$s^{u_0}(x_{i+k}) \geq s^{u_0}(x_i) + \sum_{j=i}^{i+k-1} C^{u_0}(x_j) \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right). \quad (4.2)$$

Our following analysis in Section 4.3 and Section 4.4 focuses on determining how interference may induce a deviation from the FIFO schedule described by Property 4.1.

4.3 The Start of an Eligible Interval

In this section, we study the stream of an arbitrary priority class X during an arbitrary execution u , which may encounter any type of interference other than the priority class X , i.e., \mathbb{F}^u may also contain frames that are not from X . We show that the start of an Eligible Interval of X under such interference will always be earlier than or equal to the uninterfered start time of the first transmitted frame of X of that Eligible Interval.

Due to interference, transmissions of frames in X may cluster, and multiple X frames can be transmitted in a single Eligible Interval, see Figure 4.2. Although the Eligible Intervals of the interfered execution are quite different compared to the uninterfered execution, we manage to relate the start times of the Eligible Intervals in the interfered execution u to the start times of frames in the uninterfered execution u_0 .

Obviously, for the uninterfered execution u_0 the analysis in the previous section may be used, which gives us the start times $s^{u_0}(x)$ of the uninterfered execution.

Now, consider a single Eligible Interval E of X in the execution u , and let $x_1 \in \mathbb{F}^u \cap X$ be the first frame from X transmitted during E , i.e., the first frame such that $s^u(x_1) \in E$. We aim to prove that the start of an Eligible Interval in u always lies before or at the transmission of the first frame x_1 in u_0 , i.e., $E_s \leq s^{u_0}(x_1)$.

Theorem 4.1. *Let x be the first frame of X transmitted in an Eligible Interval $E = [E_s, E_e)$ of X in execution u , and let u_0 denote the uninterfered execution of X associated with u , then $E_s \leq s^{u_0}(x)$.*

Proof. The proof proceeds by induction on the sequence of Eligible Intervals, which we denote by E^j for $j \geq 1$.

Clearly, the first Eligible Interval, E^1 , starts as soon as there is pending load, i.e., at $E_s^1 = a^u(x_1)$ with x_1 the first frame of X . Incidentally, this

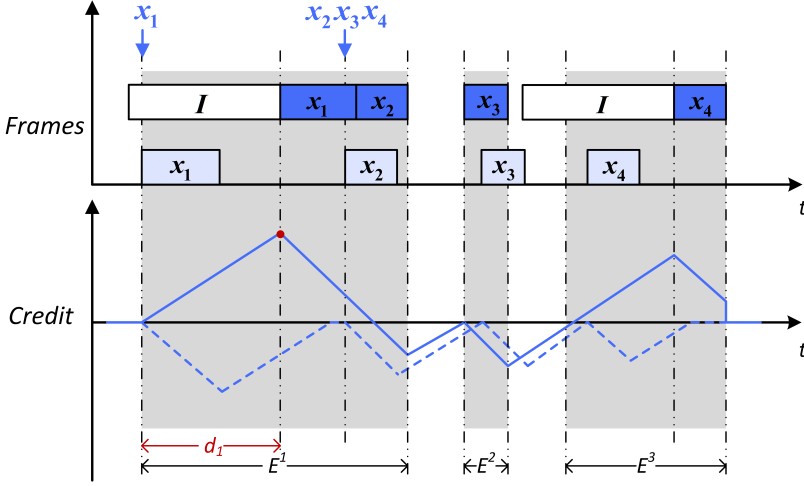


Figure 4.2: The transmission of a stream X with the presence of inter-priority interference. Rectangles labeled with I represent the transmission of interference. Solid rectangles represent the transmission of stream X , while shaded ones represent the interference-free transmission of X . Similarly, solid lines represent the credit evolution of X under interference, while dashed lines represent the credit evolution without interference. The Eligible Interval E^1 starts at the start time of its first frame x_1 in the uninterfered execution u_0 , i.e., $E_s^1 = s^{u_0}(x_1)$. For E^2 and E^3 , the Eligible Interval starts before the start time of its first frame in the uninterfered execution, i.e., $E_s^2 < s^{u_0}(x_3)$ and $E_s^3 < s^{u_0}(x_4)$, see Theorem 4.1. The relative delay of x_1 , labeled with d_1 , is proportional to the maximum credit achieved, marked using a dot: $s^u(x_1) - s^{u_0}(x_1) \leq \frac{CR_X^u(s^u(x_1))}{\alpha_X^+}$, see Theorem 4.2.

is also the start time of x_1 in the uninterfered schedule. The first frame can start as soon as it arrives, i.e., $a^{u_0}(x_1) = s^{u_0}(x_1)$. By Definition 3.1, $a^u(x_1) = a^{u_0}(x_1)$, so we have $E_s^1 = s^{u_0}(x_1)$, thus proving the base case.

Next, consider that x_k is the first frame of X in E^j and that frames x_k, \dots, x_l of X are transmitted during E^j and assume that this theorem holds for E^j , i.e., $E_s^j \leq s^{u_0}(x_k)$. Now we need to show $E_s^{j+1} \leq s^{u_0}(x_{l+1})$, with x_{l+1} the first frame of X in E^{j+1} . We know by definition of Eligible Interval that at the start of the interval E_s^j , the credit equals $CR_X^u(E_s^j) = 0$. Because by definition there is load pending during E^j , credit will fall during the transmission of those frames, and any remaining time in E^j credit will

rise, see Figure 4.3. At the end of the interval E_e^j we therefore find:

$$\begin{aligned} CR_X^u(E_e^j) \\ = \alpha_X^+ \cdot (E_e^j - E_s^j - \sum_{i=k}^l C^u(x_i)) - \alpha_X^- \cdot \sum_{i=k}^l C^u(x_i) . \end{aligned} \quad (4.3)$$

Now, if this value is positive, it must be reset to zero because if the credit is positive, there cannot be a pending load at the end of an Eligible Interval. In that case the start of E^{j+1} lies at the arrival of frame x_{l+1} , i.e., $E_s^{j+1} = a^u(x_{l+1})$. Furthermore, we know that $s^{u_0}(x_{l+1}) \geq a^{u_0}(x_{l+1}) = a^u(x_{l+1}) = E_s^{j+1}$, which concludes this case.

In case the value $CR_X^u(E_e^j)$ is negative, we know that E^{j+1} can only start after credit has recovered to zero, which happens at:

$$t_R = E_e^j + CR_X^u(E_e^j)/\alpha_X^+ = E_s^j + \sum_{i=k}^l C^u(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}). \quad (4.4)$$

Note that the credit recovery time t_R only depends on the start time of E^j and the total (inflated) transmission of class X within E^j .

Using the induction hypothesis $E_s^j \leq s^{u_0}(x_k)$ and the definition of uninterfered execution (Definition 3.1), we can now calculate:

$$\begin{aligned} E_s^{j+1} &= t_R \max a^u(x_{l+1}) \\ &= \left(E_s^j + \sum_{i=k}^l C^u(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) \right) \max a^u(x_{l+1}) \\ &\leq \left(s^{u_0}(x_k) + \sum_{i=k}^l C^u(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) \right) \max a^u(x_{l+1}) \\ &= \left(s^{u_0}(x_k) + \sum_{i=k}^l C^{u_0}(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) \right) \max a^{u_0}(x_{l+1}) \end{aligned} \quad (4.5)$$

and further using Corollary 4.1

$$E_s^{j+1} \leq s^{u_0}(x_{l+1}) \max a^{u_0}(x_{l+1}) \leq s^{u_0}(x_{l+1}) , \quad (4.6)$$

which concludes the proof. \square

This theorem allows us further to relate the delay experienced by any frame to the uninterfered execution by just studying the interference within an Eligible Interval, which is addressed in the next section.

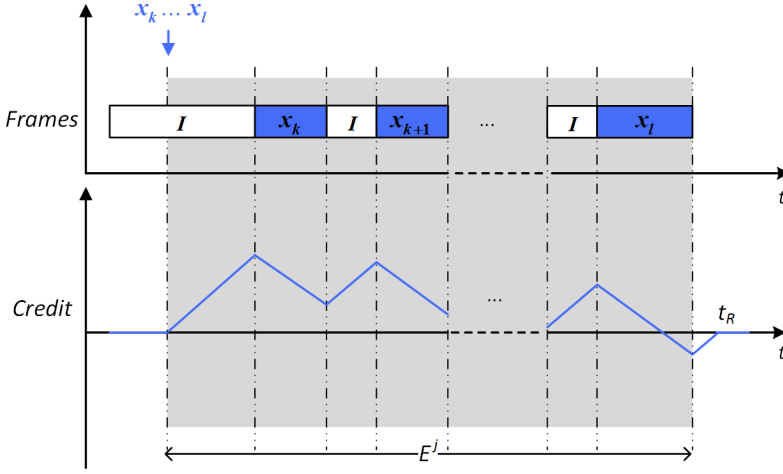


Figure 4.3: The transmission of a stream X with the presence of inter-priority interference in an Eligible Interval E^j . The rectangles labeled with I represent the transmission of interference, while the remaining rectangles represent the transmission of $x_k \dots x_l$. The credit of X decreases while frames from X are transmitting while the credit increases for the remaining time in E . In this example plot, the Eligible Interval ends due to negative credit. This credit will recover to zero at t_R .

4.4 Credit Represents Delay

Now that we have an upper bound on the start time of each Eligible Interval, we can study the relative delay of frames transmitted within an Eligible Interval. As it turns out, for those frames, the relative delay compared to the uninterfered execution is proportional to the credit of the shaper at the start of transmission in the interfered execution, which gives us the second main theorem from Cao et al. (2016b), see Figure 4.2.

Theorem 4.2. *Given a stream X and some execution u , subject to interference, and its associated uninterfered execution u_0 . For each frame $x \in \mathbb{F}^u \cap X$, we find:*

$$s^u(x) \leq s^{u_0}(x) + \frac{CR_X^u(s^u(x))}{\alpha_X^+}. \quad (4.7)$$

Proof. The proof is with induction on the sequence of frames $x_k \dots x_l$ in an Eligible Interval E_s . For the first frame x_k we know using the previous theorem that $E_s \leq s^{u_0}(x_k)$. Furthermore, at E_s credit is 0, and there is pending load in an Eligible Interval, so until $s^u(x_k)$ credit will be rising

with the rate α_X^+ . As a consequence, we find

$$s^u(x_k) = E_s + \frac{CR_X^u(s^u(x_k)) - CR_X^u(E_s)}{\alpha_X^+} \leq s^{u_0}(x_k) + \frac{CR_X^u(s^u(x_k))}{\alpha_X^+}. \quad (4.8)$$

Next, assume that the relation in Eq. (4.8) holds for some frame x_j with $k \leq j < l$, then we have the Iq. (4.9) as the induction hypothesis. Note that we use the abbreviation Iq. to represent an inequality throughout this dissertation.

$$s^u(x_j) \leq s^{u_0}(x_j) + \frac{CR_X^u(s^u(x_j))}{\alpha_X^+}. \quad (4.9)$$

We find for x_{j+1} that credit falls during the transmission of x_j and then rises until the start of x_{j+1} because in an Eligible Interval there is always pending load, so credit at the start of x_{j+1} equals (see Figure 4.3):

$$\begin{aligned} CR_X^u(s^u(x_{j+1})) &= CR_X^u(s^u(x_j)) - \alpha_X^- \cdot C^u(x_j) \\ &\quad + \alpha_X^+ \cdot (s^u(x_{j+1}) - s^u(x_j) - C^u(x_j)) \end{aligned} \quad (4.10)$$

Furthermore, we know from Corollary 4.1 that

$$s^{u_0}(x_{j+1}) \geq s^{u_0}(x_j) + C^{u_0}(x_j) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) \quad (4.11)$$

and so we obtain, using the induction hypothesis (Iq. (4.9)) in the second step,

$$\begin{aligned} &s^u(x_{j+1}) \\ &\quad \{\text{see Eq. (4.10)}\} \\ &= s^u(x_j) + C^u(x_j) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) + \frac{CR_X^u(s^u(x_{j+1}))}{\alpha_X^+} - \frac{CR_X^u(s^u(x_j))}{\alpha_X^+} \\ &\leq s^u(x_j) + C^u(x_j) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) + \frac{CR_X^u(s^u(x_{j+1}))}{\alpha_X^+} - (s^u(x_j) - s^{u_0}(x_j)) \\ &= s^{u_0}(x_j) + C^u(x_j) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) + \frac{CR_X^u(s^u(x_{j+1}))}{\alpha_X^+} \\ &= s^{u_0}(x_j) + C^{u_0}(x_j) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}) + \frac{CR_X^u(s^u(x_{j+1}))}{\alpha_X^+} \\ &\leq s^{u_0}(x_{j+1}) + \frac{CR_X^u(s^u(x_{j+1}))}{\alpha_X^+}, \end{aligned} \quad (4.12)$$

which concludes the proof. \square

Now, to prove a bound on the delay in a given execution, we only need to prove a bound on the maximum credit that can be built up in that execution. As the composition of high- and low-priority interfering traffic does influence the maximum attainable credit, we explain the calculation of the maximum credit with respect to the assumed compositions in the next chapter.

To sum up, we introduce the definition of Eligible Interval tailored for idling servers and its instantiation for the Credit-Based Shaper. Furthermore, we invent a new methodology, the relative analysis, for the worst-case response time analysis of Credit-Based Shaping. This is achieved by firstly studying the uninterfered execution and then comparing an interfered execution to its uninterfered one. Finally, this comparison leads to two essential theorems. The first one (Theorem 4.1) gives an upper bound on the start times of Eligible Intervals, which transforms the global timing analysis into a local one, implying that focusing the analysis on a single Eligible Interval is sufficient. The second one (Theorem 4.2) translates the problem of finding the maximum delay into the problem of finding the maximum credit. It is shown in the next chapter how these two theorems greatly simplify further analysis. Note that we do not restrict the interference to any particular type. Hence, these two theorems are not merely applicable for this dissertation, where we consider the interference as the union of all high-priority frames under Credit-Based Shaping and all low-priority frames, but also for more complicated combinations of interference.

Chapter 5

Relative Worst-Case Response Time Analysis

Our analysis focuses on the behavior at the output port of an Ethernet TSN switch, as depicted in Figure 3.1. This means we assume that incoming frames have already been routed to the proper output port and that the switch can freely transmit at its output. At the output port, frames are queued according to their priorities, resulting in several streams of frames, one for each priority. We keep track of when frames of a certain stream arrive and when they are transmitted, and we calculate how long a frame may remain in the output buffer. In other words, we are interested in the maximum amount of time between a frame entering the buffer and finishing transmission.

Unlike the conventional methods (Bordoloi et al. (2014); Diemer et al. (2012b)) that need detailed models of all relevant traffic and calculate the worst-case response time of the traffic of interest in one attempt, we propose a relative approach to first compute the worst-case response time in an uninterfered execution before calculating the worst-case response time in an interfered execution. The challenge with this approach lies in investigating whether there exists an upper bound for the delay on the start time of any frame due to the introduction of inter-priority interference. We give rigorous proof in this chapter to establish such an upper bound. The associated expression for this upper bound of the relative delay is simple. It only takes the maximum frame sizes and bandwidth reservations of the interfering traffic into account, i.e., the knowledge on the shaper level. Hence, it leads to an independent analysis, for which the detailed traffic model of the inter-priority interference is irrelevant.

It is noted that this chapter includes a recapitulation of the analysis presented in our publication Cao et al. (2018b). We consider a generic setup, i.e., the interference consisting of multiple high-priorities under Credit-Based Shaping and multiple low-priorities, which generalizes the analysis in our previous works Cao et al. (2016b,a) and makes the analysis more widely applicable.

In this chapter, we first prove a theorem to derive the minimum sum of credits of a set of Credit-Based Shapers in Section 5.1. To put this as the first step is because the minimum sum of credits of a set of high-priority Credit-Based Shapers turns out to be essential to obtain the maximum attainable credit of the priority class of interest. Then, we derive in Section 5.2 the upper bound of relative delay, i.e., the worst-case relative delay, by bounding the maximum attainable credit. Furthermore, we show in Section 5.3 that the estimate of the worst-case relative delay in a simpler setup of interfering traffic is consistent with the estimate given by the generic solution. In the end, we provide a concrete example with the presence of multiple high-priority shapers in Section 5.4 to show how to use the proven theorems to derive the worst-case relative delay in practice.

5.1 Calculating the Minimum Credit

From Theorem 4.2 introduced in Chapter 4, it is straightforward to conclude that we can bound the relative delay, i.e., the delay on the start time of any frame in the interfered execution compared to the uninterfered one, by bounding the maximum credit that can be built up by a shaper.

To calculate this maximum credit, we need to estimate the amount of interference that can occur, since positive credit of X only builds up while pending frames of X wait for eligible transmissions from other priorities to finish. Given the interference consisting of a set of high-priority Credit-Based Shapers, it turns out to be essential to estimate the minimum credit of the set. We define the minimum credit of a set of Credit-Based Shapers \mathbb{X} as:

$$CR_{\mathbb{X}}^{\min} \stackrel{\text{def}}{=} \inf_{u \in \mathbb{U}} \inf_{t \in \mathbb{R}_0^+} \sum_{X \in \mathbb{X}} CR_X^u(t). \quad (5.1)$$

and we have $CR_{\emptyset}^{\min} = 0$ as the sum of the elements of an empty set is zero.

Now we consider the total credit $CR_{\mathbb{X}}^u$ of a set of priority classes $\mathbb{X} \subseteq \mathbb{P}$; we could, of course, simply add up the minimum credits of each of its

elements. However, for sets larger than 1 element, there is no execution in which this sum can be actually reached, because as the transmission from one of the elements of \mathbb{X} is proceeding towards its minimum, the credits of other elements in \mathbb{X} will be recovering. In the following theorem, we prove a method with which a tight bound of the minimum credit of a set \mathbb{X} can be iteratively achieved. It is important to assume that the total reservation of the set $\alpha_{\mathbb{X}}^+$ does not exceed the total bandwidth BW . This is a valid assumption in practice.

Theorem 5.1. *Given a finite non-empty set of priority classes \mathbb{X} under Credit-Based Shaping with $\alpha_{\mathbb{X}}^+ \leq BW$, we have*

$$CR_{\mathbb{X}}^{\min} = - \sup_{X \in \mathbb{X}} (\alpha_{\mathbb{X}}^- \cdot C_X^{\max} - CR_{\mathbb{X} \setminus X}^{\min}). \quad (5.2)$$

Proof. First we prove that for all executions u and times t we have $CR_{\mathbb{X}}^u(t) \geq - \sup_{X \in \mathbb{X}} (\alpha_{\mathbb{X}}^- \cdot C_X^{\max} - CR_{\mathbb{X} \setminus X}^{\min})$ by contradiction of the opposite. Assume that there exists an execution v and time t such that $CR_{\mathbb{X}}^v(t) < - \sup_{X \in \mathbb{X}} (\alpha_{\mathbb{X}}^- \cdot C_X^{\max} - CR_{\mathbb{X} \setminus X}^{\min})$. Then, given the condition $\alpha_{\mathbb{X}}^+ \leq BW$, for each individual $X \in \mathbb{X}$, we have:

$$\begin{aligned} CR_X^v(t) &= CR_{\mathbb{X}}^v(t) - CR_{\mathbb{X} \setminus X}^v(t) \\ &\quad \{\text{by assumption}\} \\ &< - \sup_{X' \in \mathbb{X}} (\alpha_{\mathbb{X}}^- \cdot C_{X'}^{\max} - CR_{\mathbb{X} \setminus X'}^{\min}) - CR_{\mathbb{X} \setminus X}^v(t) \\ &\leq - \alpha_{\mathbb{X}}^- \cdot C_X^{\max} + CR_{\mathbb{X} \setminus X}^{\min} - CR_{\mathbb{X} \setminus X}^v(t) \\ &\quad \{\text{by definition of minimum credit}\} \\ &\leq - \alpha_{\mathbb{X}}^- \cdot C_X^{\max} \\ &= - (BW - \alpha_{\mathbb{X}}^+) \cdot C_X^{\max} \\ &\leq 0. \end{aligned} \quad (5.3)$$

In other words, for each individual X , we have negative credit at time t . From the shaping rules, we know that this is only possible if all shapers are currently either transmitting or in recovery. We reason that credit at an earlier point would even be lower if all shapers are in recovery at t , and therefore we may assume without loss of generality that at time t one of the shapers is transmitting. Let $x \in X$ be the frame that is in transmission at t , and recall that all others are recovering. Furthermore, recall that at the start time of this transmission we have $CR_X^v(s^v(x)) \geq 0$, so we may derive:

$$\begin{aligned}
& CR_{\mathbb{X} \setminus X}^v(s^v(x)) \\
&= CR_{\mathbb{X}}^v(s^v(x)) - CR_X^v(s^v(x)) \\
&\leq CR_{\mathbb{X}}^v(s^v(x)) \\
&= CR_{\mathbb{X}}^v(t) - \alpha_{\mathbb{X} \setminus X}^+ \cdot (t - s^v(x)) + \alpha_X^- \cdot (t - s^v(x)) \\
&= CR_{\mathbb{X}}^v(t) - \alpha_{\mathbb{X} \setminus X}^+ \cdot (t - s^v(x)) + (BW - \alpha_X^+) \cdot (t - s^v(x)) \\
&= CR_{\mathbb{X}}^v(t) + (BW - \alpha_X^+) \cdot (t - s^v(x)) \\
&= CR_{\mathbb{X}}^v(t) + \alpha_X^- \cdot (t - s^v(x)) \tag{5.4} \\
&\quad \{\text{by maximum frame size}\} \\
&\leq CR_{\mathbb{X}}^v(t) + \alpha_X^- \cdot C_X^{\max} \\
&\quad \{\text{by assumption}\} \\
&< - \sup_{X' \in \mathbb{X}} (\alpha_X^- \cdot C_{X'}^{\max} - CR_{\mathbb{X} \setminus X'}^{\min}) + \alpha_X^- \cdot C_X^{\max} \\
&\leq - \alpha_X^- \cdot C_X^{\max} + CR_{\mathbb{X} \setminus X}^{\min} + \alpha_X^- \cdot C_X^{\max} \\
&= CR_{\mathbb{X} \setminus X}^{\min},
\end{aligned}$$

which contradicts the definition of the minimum credit, and therefore proves that our estimate is sound, i.e., $CR_{\mathbb{X}}^u(t) \geq - \sup_{X \in \mathbb{X}} (\alpha_X^- \cdot C_X^{\max} - CR_{\mathbb{X} \setminus X}^{\min})$.

Secondly, to prove that our estimate is tight, we recursively construct an execution in which this estimate is actually achieved. The base case is trivial. For the empty set, we have $CR_{\emptyset}^{\min} = 0$. This is achieved by an execution u_{\emptyset} in which nothing is transmitted. For the recursive case, consider a set \mathbb{X} containing at least 1 element. We first determine which $X \in \mathbb{X}$ achieves the supremum in the formula $- \sup_{X \in \mathbb{X}} (\alpha_X^- \cdot C_X^{\max} - CR_{\mathbb{X} \setminus X}^{\min})$. By construction, we already have an execution $u_{\mathbb{X} \setminus X}$ that achieves the minimum credit of $\mathbb{X} \setminus X$. Since it is not necessary to transmit a frame from X in the execution $u_{\mathbb{X} \setminus X}$, we further assume that there is no transmission and pending frames from X so that the credit of X remains 0 in $u_{\mathbb{X} \setminus X}$. Then, we create $u_{\mathbb{X}}$ by appending a single transmission of a maximal frame from X to the execution $u_{\mathbb{X} \setminus X}$, choosing the arrival time of the frame equal to the end of the last transmission in $u_{\mathbb{X} \setminus X}$. By such a construction, the credit of \mathbb{X} reaches $-(\alpha_X^- \cdot C_X^{\max} - CR_{\mathbb{X} \setminus X}^{\min})$ at the end of the execution $u_{\mathbb{X}}$. Hence, the estimate of the minimum credit is tight, which concludes the proof. \square

Note that the original theorem regarding the single priority, see Theorem 3 in Cao et al. (2018b), directly follows as a corollary.

Corollary 5.1. *For a single priority class $X \in \mathbb{X}$ under Credit-Based Shaping with $\alpha_X^+ \leq BW$, we find $CR_X^{\min} = -\alpha_X^- \cdot C_X^{\max}$.*

The method for computing the minimum credit of a set of Credit-Based Shapers has just been presented. Note that the complexity of this computation is factorial. For a set of shapers \mathbb{X} containing N priority classes, there exists an order for these N priority classes to reach the minimum credit. To cover all the possibilities of order, it needs $N \cdot (N - 1) \cdots 2 \cdot 1 = N!$ calculations.

5.2 Bounding the Maximum Relative Delay

We estimate the maximum relative delay for a priority class of interest by adding all possible low- and high-priority interference. The relative delay caused by the low-priority interference is trivial. For high-priority interference consisting of a set of Credit-Based Shapers, the total credit of the set should drop to its minimum at its slowest decreasing rate to prevent transmission from the priority class of interest as much as possible. In this section, we first introduce the slowest decreasing rate of the total credit of a set of Credit-Based Shapers. Then, together with the minimum credit of a set of Credit-Based Shapers (see Section 5.1), we introduce how to obtain the maximum relative delay for a priority class of interest.

Theorem 5.2. *Given a set of priority classes \mathbb{X} under Credit-Based Shaping with $\alpha_{\mathbb{X}}^+ \leq BW$, if during an execution u at every point t' in an interval $[t, t']$ there is a frame from \mathbb{X} transmitting, then the total credit of \mathbb{X} decreases with at least a rate $\alpha_{\mathbb{X}}^-$, i.e.,*

$$CR_{\mathbb{X}}^u(t') \leq CR_{\mathbb{X}}^u(t) - \alpha_{\mathbb{X}}^- \cdot (t' - t). \quad (5.5)$$

Proof. The slowest possible decrease of total credit is achieved, obviously, when credit for all shapers in \mathbb{X} is increasing, except for the shaper that is currently transmitting. Regardless of whether the shapers are increasing because they are recovering or have a pending load, assume without loss of generality that shaper $X \in \mathbb{X}$ is transmitting and find:

$$\begin{aligned} & CR_{\mathbb{X}}^u(t') \\ & \leq CR_{\mathbb{X}}^u(t) + \alpha_{\mathbb{X} \setminus X}^+ \cdot (t' - t) - \alpha_X^- \cdot (t' - t) \\ & = CR_{\mathbb{X}}^u(t) + \alpha_{\mathbb{X} \setminus X}^+ \cdot (t' - t) - (BW - \alpha_X^+) \cdot (t' - t) \\ & = CR_{\mathbb{X}}^u(t) + \alpha_{\mathbb{X} \setminus X}^+ \cdot (t' - t) - BW \cdot (t' - t) + \alpha_X^+ \cdot (t' - t) \\ & = CR_{\mathbb{X}}^u(t) + (\alpha_{\mathbb{X} \setminus X}^+ + \alpha_X^+ - BW) \cdot (t' - t) \\ & = CR_{\mathbb{X}}^u(t) - \alpha_{\mathbb{X}}^- \cdot (t' - t), \end{aligned} \quad (5.6)$$

which concludes the proof. \square

Theorem 5.3. *Given a set of priority classes \mathbb{X} under Credit-Based Shaping with $\alpha_{\mathbb{X}}^+ \leq BW$, if during an execution u at every point t' in an interval $[t, t']$ all shapers from \mathbb{X} either have pending load or negative credit, then the total credit of \mathbb{X} decreases with exactly a rate $\alpha_{\mathbb{X}}^-$, i.e.,*

$$CR_{\mathbb{X}}^u(t') = CR_{\mathbb{X}}^u(t) - \alpha_{\mathbb{X}}^- \cdot (t' - t). \quad (5.7)$$

Proof. The assumption that all shapers in \mathbb{X} either have pending load or negative credit ensures that all other shapers' credits increase when one shaper is transmitting. Thus, we have:

$$\begin{aligned} CR_{\mathbb{X}}^u(t') &= CR_{\mathbb{X}}^u(t) + \alpha_{\mathbb{X} \setminus X}^+ \cdot (t' - t) - \alpha_X^- \cdot (t' - t) \\ &= CR_{\mathbb{X}}^u(t) - \alpha_{\mathbb{X}}^- \cdot (t' - t), \end{aligned} \quad (5.8)$$

and the slowest decreasing rate is attained. \square

Finally, combining the minimum credit calculation in Theorem 5.1 and the slowest decreasing rate in Theorem 5.2, we estimate the maximum credit of a priority class X by adding the possible interference from low- and high-priority classes. From low-priority classes, only a single frame can interfere, which starts non-preemptively 'just before' the arrival of a frame from X and leads to an initial credit increase of X . This low-priority frame also leads to an initial increase in the total credit of high-priority classes. The maximum credit of X can be obtained when the high-priority shapers continue transmission before the total credit decreases from the initial increase to its minimum at the slowest possible rate.

The following theorem gives the upper bound of the relative delay, i.e., $\sup_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x))$ in the relative worst-case response time calculation (see Eq. (3.4)) by bounding the maximum credit. We define the maximum credit of a Credit-Based Shaper X as:

$$CR_X^{\max} \stackrel{\text{def}}{=} \sup_{u \in \mathbb{U}} \sup_{t \in \mathbb{R}_0^+} CR_X^u(t). \quad (5.9)$$

This bound of relative delay, notably, is not necessarily tight because the execution that leads to the slowest possible decrease in credit is not always also an execution that leads to minimum credit, which is covered in Chapter 6.

Theorem 5.4. *Given a particular stream of interest associated with priority class $M \in \mathbb{P}$. Consider the set $\mathbb{H} = \{X \in \mathbb{P} \mid X > M\}$ of all high-priority*

classes, and the set $\mathbb{L} = \{X \in \mathbb{P} \mid X < M\}$ of all low-priority classes. Assume that M and \mathbb{H} are under Credit-Based Shaping, and $\alpha_{\mathbb{H}}^+ + \alpha_M^+ \leq BW$. Then, for any execution u with associated uninterfered execution u_0 and any frame $m \in M$ we find:

$$s^u(m) - s^{u_0}(m) \leq C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-}\right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-}. \quad (5.10)$$

Proof. To prove this theorem, we only need to show that during any execution u , the credit $CR_M^u(t)$ is bounded by

$$\alpha_M^+ \cdot \left(C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-}\right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} \right), \quad (5.11)$$

and subsequently use Theorem 4.2 to convert the credit bound into the latency bound. Next, to study the variation of the credit $CR_M^u(t)$ over time, we divide time in the execution u into phases of three exclusive types depending on what is in transmission: idling phase (nothing is in transmission within this interval), L phase (only \mathbb{L} frames are in consecutive transmission) and MH phase (only frames from M or \mathbb{H} are in consecutive transmission). Since these three conditions are mutually exclusive and cover all possibilities, a phase ends when another type of phase takes over.

Idling Phase

When nothing is in transmission, either the credit of every priority class in \mathbb{H} and the credit of M is negative (transmission is not allowed by the shaper), or there is no pending load (positive credit is reset to 0). Hence, at any time t in an idling phase, the credit of every stream $X \in \mathbb{H} \cup \{M\}$ is less than or equal to 0:

$$CR_X^u(t) \leq 0. \quad (5.12)$$

L Phase

An L frame cannot start transmission when either the credit of M or the credit of some $H \in \mathbb{H}$ is strictly positive (because this implies pending load of a high-priority class). Therefore, we find those credits are 0 or negative at the start of any L transmission. During the transmission, credit of $X \in \mathbb{H} \cup \{M\}$ rises with a rate at most α_X^+ . Credit of $\mathbb{H} \cup \{M\}$ only rises above 0 if there are frames pending, so the increase may be less. Once credit of $\mathbb{H} \cup \{M\}$ rises strictly above 0 no new L transmission can start, but the current one can finish. As a consequence, for any $X \in \mathbb{H} \cup \{M\}$ we

find credit at a time t in an L phase is bounded by the transmission of a single L frame of maximum size:

$$CR_X^u(t) \leq C_{\mathbb{L}}^{\max} \cdot \alpha_X^+. \quad (5.13)$$

MH Phase

The credit variation is more complex in an MH phase. Let t_s denote the start time of a given MH phase. To bound the credit of M at any time t ($t \geq t_s$) during this phase, we study for each $X \in \mathbb{H} \cup \{M\}$ the cumulative time during which frames in X are transmitted in $[t_s, t]$. We denote this by writing δ_X for any $X \in \mathbb{H} \cup \{M\}$ and know that $\sum_{X \in \mathbb{H} \cup \{M\}} \delta_X = t - t_s$. Our subsequent reasoning is mainly based on the credit variation rules and the reservation bound $\alpha_{\mathbb{H}}^+ + \alpha_M^+ \leq BW$.

Clearly, in an MH phase, there is always a transmission from $\mathbb{H} \cup \{M\}$ going on. Therefore, the credit of some $X \in \mathbb{H} \cup \{M\}$ will rise as long as it is not transmitting itself, and as long as it has either negative credit or pending load. If there is no pending load, credit will not rise above 0. In this way, the total credit of \mathbb{H} rises with a rate at most $\alpha_{\mathbb{H}}^+$ while M is transmitting, i.e., within the cumulative duration δ_M .

Given Theorem 5.2, the total credit of \mathbb{H} drops with a rate at least $\alpha_{\mathbb{H}}^-$ when one of the priorities in \mathbb{H} is transmitting within the cumulative duration $\sum_{H \in \mathbb{H}} \delta_H$. Therefore, we have the total credit of the high priorities as:

$$CR_{\mathbb{H}}^u(t) \leq CR_{\mathbb{H}}^u(t_s) + \alpha_{\mathbb{H}}^+ \cdot \delta_M - \alpha_{\mathbb{H}}^- \cdot \sum_{H \in \mathbb{H}} \delta_H. \quad (5.14)$$

Note that this upper bound is reached only if throughout the interval $[t_s, t]$, when one stream is in transmission, all other streams keep increasing their credits.

Also, we know by the definition of minimum credit (see Eq. (5.1)) that the total credit of the high priorities has a lower bound of $CR_{\mathbb{H}}^{\min}$. Therefore, we obtain:

$$CR_{\mathbb{H}}^{\min} \leq CR_{\mathbb{H}}^u(t) \leq CR_{\mathbb{H}}^u(t_s) + \alpha_{\mathbb{H}}^+ \cdot \delta_M - \alpha_{\mathbb{H}}^- \cdot \sum_{H \in \mathbb{H}} \delta_H. \quad (5.15)$$

We further use the definition $\alpha_X^- = BW - \alpha_X^+$ to derive that:

$$\begin{aligned}
\alpha_{\mathbb{H}}^+ + \alpha_M^+ \leq BW &\iff \begin{aligned} \alpha_{\mathbb{H}}^+ &\leq \alpha_M^- \\ \alpha_M^+ &\leq \alpha_{\mathbb{H}}^- \end{aligned} \\
&\iff \alpha_{\mathbb{H}}^+ \cdot \alpha_M^+ \leq \alpha_{\mathbb{H}}^- \cdot \alpha_M^- \\
&\iff \frac{\alpha_M^+ \cdot \alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-} - \alpha_M^- \leq 0.
\end{aligned} \tag{5.16}$$

Next, we obtain the upper bound on the credit for M in a similar way as on the total credit for the set \mathbb{H} and further refine this bound using Iq. (5.15) and Iq. (5.16) as follows:

$$\begin{aligned}
&CR_M^u(t) \\
&\{\text{according to credit evolution rules}\} \\
&\leq CR_M^u(t_s) + \alpha_M^+ \cdot \sum_{H \in \mathbb{H}} \delta_H - \alpha_M^- \cdot \delta_M \\
&\{\text{see Iq. (5.15)}\} \\
&\leq CR_M^u(t_s) + \frac{\alpha_M^+}{\alpha_{\mathbb{H}}^-} \cdot (CR_{\mathbb{H}}^u(t_s) + \alpha_{\mathbb{H}}^+ \cdot \delta_M - CR_{\mathbb{H}}^{\min}) - \alpha_M^- \cdot \delta_M \tag{5.17} \\
&= CR_M^u(t_s) + \frac{\alpha_M^+}{\alpha_{\mathbb{H}}^-} \cdot (CR_{\mathbb{H}}^u(t_s) - CR_{\mathbb{H}}^{\min}) + \left(\frac{\alpha_M^+ \cdot \alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-} - \alpha_M^- \right) \cdot \delta_M \\
&\{\text{see Iq. (5.16)}\} \\
&\leq CR_M^u(t_s) + \frac{\alpha_M^+}{\alpha_{\mathbb{H}}^-} \cdot (CR_{\mathbb{H}}^u(t_s) - CR_{\mathbb{H}}^{\min}).
\end{aligned}$$

Finally, an MH phase can only follow an idling phase or an L phase. Therefore, the initial credits $CR_M^u(t_s)$ are bounded by the upper bounds obtained in these two phases, see Iq. (5.12) and Iq. (5.13). The credit bound in an L phase is higher. To reach the maximum credit for M in an MH phase, that MH phase must be preceded by an L phase with a frame of the maximum size and we derive:

$$\begin{aligned}
&CR_M^u(t) \\
&\leq CR_M^u(t_s) + \frac{\alpha_M^+}{\alpha_{\mathbb{H}}^-} \cdot (CR_{\mathbb{H}}^u(t_s) - CR_{\mathbb{H}}^{\min}). \\
&\{\text{see Iq. (5.13)}\} \\
&\leq C_{\mathbb{L}}^{\max} \cdot \alpha_M^+ + \frac{\alpha_M^+}{\alpha_{\mathbb{H}}^-} \cdot (C_{\mathbb{L}}^{\max} \cdot \alpha_{\mathbb{H}}^+ - CR_{\mathbb{H}}^{\min}) \\
&= \alpha_M^+ \cdot \left(C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-} \right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} \right).
\end{aligned} \tag{5.18}$$

Combining the upper bounds of credit for M in all three possible phases: Iq. (5.12) for idling phases, Iq. (5.13) for L phases, and Iq. (5.18) for MH phases, we can derive the upper bound for the maximum credit for M (as defined in Eq. (5.9)):

$$CR_M^{\max} \leq \alpha_M^+ \cdot \left(C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-} \right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} \right). \quad (5.19)$$

As a final step, we apply Theorem 4.2 and derive:

$$\begin{aligned} s^u(m) - s^{u_0}(m) &\leq \frac{CR_M^u(s^u(m))}{\alpha_M^+} \\ &\leq \frac{CR_M^{\max}}{\alpha_M^+} \\ &\quad \{\text{see Iq. (5.19)}\} \\ &\leq \frac{\alpha_M^+ \cdot \left(C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-} \right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} \right)}{\alpha_M^+} \\ &= C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-} \right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-}, \end{aligned} \quad (5.20)$$

which concludes our proof. \square

5.3 Consistency with Previous Works

We started our relative worst-case response time analysis in a much simpler scenario, i.e., one low priority or one high priority in Cao et al. (2016b) and then one low priority combined with one high priority in Cao et al. (2016a) before we resolved the generic scenario with multiple low- and high-priority interference. In this section, we recapitulate those essential theorems that we prove in our earlier papers as corollaries of Theorem 5.4 to show this consistency.

5.3.1 Only Low-Priority Interference

Given only low-priority interference, it was proven in Cao et al. (2016b) (see Theorem 3) that the stream M suffers a delay of at most one L frame of the maximum size.

Corollary 5.2. *Given a particular stream of interest associated with priority class $M \in \mathbb{P}$. Consider the set $\mathbb{L} = \{X \in \mathbb{P} \mid X < M\}$ of all low-priority classes. Assume that M is under Credit-Based Shaping, and $\alpha_M^+ \leq BW$.*

Then, for any execution u with associated uninterfered execution u_0 and any frame $m \in M$ we find:

$$s^u(m) - s^{u_0}(m) \leq C_{\mathbb{L}}^{\max}. \quad (5.21)$$

Proof. In this case, the high-priority interference is excluded and thus we consider an empty set for \mathbb{H} in Theorem 5.4. Given $\alpha_{\emptyset}^+ = 0$ and $CR_{\emptyset}^{\min} = 0$, we derive:

$$\begin{aligned} s^u(m) - s^{u_0}(m) &\leq C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-}\right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} \\ &= C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\emptyset}^+}{\alpha_{\emptyset}^-}\right) - \frac{CR_{\emptyset}^{\min}}{\alpha_{\emptyset}^-} \\ &= C_{\mathbb{L}}^{\max}, \end{aligned} \quad (5.22)$$

which concludes our proof. \square

5.3.2 Only One High-Priority Interference

Given only one high-priority interference with the assumption $\alpha_H^+ + \alpha_M^+ < BW$, it was shown in Cao et al. (2016b) (see Theorem 4) that the a stream M suffers a delay by at most one H frame of the maximum size through a complicated proof. However, it is rather straightforward to derive the same conclusion with a less strict assumption $\alpha_H^+ + \alpha_M^+ \leq BW$ as the corollary of Theorem 5.4.

Corollary 5.3. *Given a particular stream of interest associated with priority class $M \in \mathbb{P}$. Consider only one high-priority class $H \in \mathbb{P}$ and $H > M$. Assume that M and H are under Credit-Based Shaping, and $\alpha_H^+ + \alpha_M^+ \leq BW$. Then, for any execution u with associated uninterfered execution u_0 and any frame $m \in M$ we find:*

$$s^u(m) - s^{u_0}(m) \leq C_H^{\max}. \quad (5.23)$$

Proof. In this case, we simply take $C_{\mathbb{L}}^{\max} = 0$, because the low-priority interference is excluded. Furthermore, the minimum credit calculation of a single high priority is simple (Corollary 5.1), $CR_{\mathbb{H}}^{\min} = CR_H^{\min} = -\alpha_H^- \cdot C_H^{\max}$.

Subsequently, we have:

$$\begin{aligned}
& s^u(m) - s^{u_0}(m) \\
& \leq C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-}\right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} \\
& = 0 \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) - \frac{CR_H^{\min}}{\alpha_H^-} \\
& \quad \{\text{see Corollary 5.1}\} \\
& = -\frac{-\alpha_H^- \cdot C_H^{\max}}{\alpha_H^-} \\
& = C_H^{\max},
\end{aligned} \tag{5.24}$$

which concludes the proof. \square

5.3.3 One High-Priority Combined with Low-Priority Interference

Similarly, we reach the same conclusion as we proved in Cao et al. (2016a) (see Theorem 3) given the combined interference.

Corollary 5.4. *Given a particular stream of interest associated with priority class $M \in \mathbb{P}$. Consider only one high-priority class $H \in \mathbb{P}$ and $H > M$, and the set $\mathbb{L} = \{X \in \mathbb{P} \mid X < M\}$ of all low-priority classes. Assume that M and H are under Credit-Based Shaping, and $\alpha_H^+ + \alpha_M^+ \leq BW$. Then, for any execution u with associated uninterfered execution u_0 and any frame $m \in M$ we find:*

$$s^u(m) - s^{u_0}(m) \leq C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{\max}. \tag{5.25}$$

Notably, this upper bound for a single high priority in Corollary 5.4 is tight since we are able to construct an execution which leads to this worst-case delay. As we mentioned before, the upper bound for the generic case with multiple high priorities, given by Theorem 5.4, is not necessarily tight. This is because the execution that leads to the slowest decreasing rate of the total credit is not always an execution that leads to the minimum total credit. However, the minimum total credit for a high-priority set consisting of a single element, i.e., $CR_H^{\min} = -\alpha_H^- \cdot C_H^{\max}$, can be always reached with the slowest descending rate of the set, i.e., $-\alpha_H^-$. The tightness of this upper bound is elaborated in more details in Chapter 6.

5.4 A Concrete Example of Multiple High-Priority Interference

Now we show how to use Theorem 5.1 and Theorem 5.4 to calculate the worst-case relative delay of stream M given a concrete set-up of the inter-priority interference as shown in Table 5.1.

Table 5.1: An example of three high-priority levels ($H_1 > H_2 > H_3$).

	α^+ Mbps	α^- Mbps	C^{\max} μs	CR^{\min} bits
H_1	10	90	3	-270
H_2	20	80	2	-160
H_3	15	85	4	-340
M	10	90	5	-450
L	N.A.	N.A.	5	N.A.

In this example, the total bandwidth BW is set to 100 Mbps. There are three high priorities, H_1 , H_2 and H_3 , in the order of priority. The total reservation of high priorities $\alpha_{\mathbb{H}}^+$ is 45 Mbps and the remaining $\alpha_{\mathbb{H}}^-$ is 55 Mbps. As α_M^+ is set to 10 Mbps, the condition $\alpha_{\mathbb{H}}^+ + \alpha_M^+ \leq BW$ is met, thus guaranteeing that the worst-case response time is finite. The maximum frame size of a priority class is given as the transmission time with a unit of μs . Note that this specific value does not represent the actual transmission time of an Ethernet frame in practice. This value is chosen in a way to simplify the calculation.

To calculate the minimum credit $CR_{\mathbb{H}}^{\min}$, we mainly use Theorem 5.1. We start with the minimum credit of a single priority, shown in Table 5.1, then with the set of two high priorities, and finally with the set of three.

Credit of the set of two priorities

$$\begin{aligned}
 & CR_{\{H_1, H_2\}}^{\min} \\
 &= -\max \left(\alpha_{\{H_1, H_2\}}^- \cdot C_{H_1}^{\max} - CR_{H_2}^{\min}, \alpha_{\{H_1, H_2\}}^- \cdot C_{H_2}^{\max} - CR_{H_1}^{\min} \right) \quad (5.26) \\
 &= -\max(370, 410) \\
 &= -410
 \end{aligned}$$

$$\begin{aligned}
& CR_{\{H_2, H_3\}}^{\min} \\
&= -\max\left(\alpha_{\{H_2, H_3\}}^- \cdot C_{H_2}^{\max} - CR_{H_3}^{\min}, \alpha_{\{H_2, H_3\}}^- \cdot C_{H_3}^{\max} - CR_{H_2}^{\min}\right) \quad (5.27) \\
&= -\max(470, 420) \\
&= -470
\end{aligned}$$

$$\begin{aligned}
& CR_{\{H_1, H_3\}}^{\min} \\
&= -\max\left(\alpha_{\{H_1, H_3\}}^- \cdot C_{H_1}^{\max} - CR_{H_3}^{\min}, \alpha_{\{H_1, H_3\}}^- \cdot C_{H_3}^{\max} - CR_{H_1}^{\min}\right) \quad (5.28) \\
&= -\max(565, 570) \\
&= -570
\end{aligned}$$

Credit of the set of three priorities

$$\begin{aligned}
& CR_{\mathbb{H}}^{\min} \\
&= -\max\left(\alpha_{\mathbb{H}}^- \cdot C_{H_1}^{\max} - CR_{\{H_2, H_3\}}^{\min}, \alpha_{\mathbb{H}}^- \cdot C_{H_2}^{\max} - CR_{\{H_1, H_3\}}^{\min}, \alpha_{\mathbb{H}}^- \cdot C_{H_3}^{\max} - CR_{\{H_1, H_2\}}^{\min}\right) \\
&= -\max(635, 680, 630) \\
&= -680
\end{aligned} \quad (5.29)$$

We construct a sequence H_1, H_3, H_2 in this example, so that the minimum credit is obtained through the calculations of $CR_{H_1}^{\min}$, $CR_{\{H_1, H_3\}}^{\min}$ and $CR_{\{H_1, H_3, H_2\}}^{\min}$ as follows:

$$\begin{aligned}
CR_{H_1}^{\min} &= -\alpha_{H_1}^- \cdot C_{H_1}^{\max} + CR_{\emptyset}^{\min} \\
CR_{\{H_1, H_3\}}^{\min} &= -\alpha_{\{H_1, H_3\}}^- \cdot C_{H_3}^{\max} + CR_{H_1}^{\min} \\
CR_{\{H_1, H_3, H_2\}}^{\min} &= -\alpha_{\{H_1, H_3, H_2\}}^- \cdot C_{H_2}^{\max} + CR_{\{H_1, H_3\}}^{\min}
\end{aligned} \quad (5.30)$$

As a next step, we use Theorem 5.4 to obtain the worst-case relative delay.

$$C_{\mathbb{L}}^{\max} \cdot \left(1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-}\right) - \frac{CR_{\mathbb{H}}^{\min}}{\alpha_{\mathbb{H}}^-} = 5 \cdot \left(1 + \frac{45}{55}\right) + \frac{680}{55} \approx 21.45 \quad (5.31)$$

We show one feasible worst-case scenario in Figure 5.1. As the total credit $CR_{\mathbb{H}}$ reaches its minimum at the slowest descent $\alpha_{\mathbb{H}}^- = 55$ Mbps, the calculated worst-case relative delay 21.45 μ s is attained. In this case, observe that the estimate is tight since there exists an execution that achieves

the worst-case relative delay. In the next chapter, tightness is extensively studied by determining sufficient conditions under which such an execution exists in general.

5.5 Conclusion

In this chapter, we have presented a relative worst-case response time analysis for a Credit-Based Shaper of interest, given the interference consisting of multiple high-priorities under Credit-Based Shaping and multiple low-priorities. Therefore, instead of calculating the worst-case response time of the stream M of interest in an interfered execution u directly, we can first compute the worst-case response time in an associated uninterfered execution u_0 and then add the worst-case relative delay.

The most important contribution of this chapter is Theorem 5.4, with which the worst-case relative delay is derived by bounding the maximum attainable credit. As a result, we can derive the worst-case relative delay in two steps: first calculate the minimum credit of a set of high-priority Credit-Based Shapers $CR_{\mathbb{H}}^{\min}$ in an iterative manner using Theorem 5.1 and secondly obtain the upper bound of the relative delay using the conclusion of Theorem 5.4. Notably, the bound of the minimum credit of a set of Credit-Based Shapers is tight, but the upper bound of the relative delay is not necessarily tight. This is because the maximum relative delay is reached only if the total credit of a set of high-priority shapers drops to its minimum at the slowest descending rate and such an execution does not always exist. The tightness of the entire approach is further discussed in the next chapter.

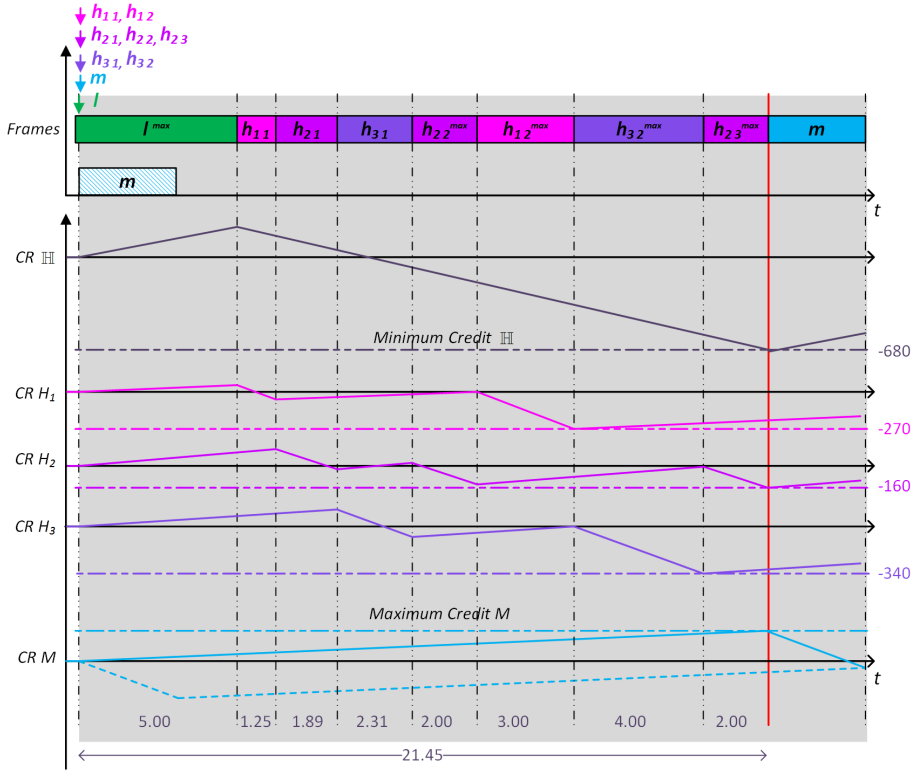


Figure 5.1: A stream of M interfered by a set of high-priority streams $H = \{H_1, H_2, H_3\}$ and a low-priority stream L as shown in Table 5.1. Arrows here indicate frame arrivals. The solid rectangles represent the transmission of streams H , M and L while the shaded ones represent the interference-free transmission of M . So do the solid and dashed lines representing the credit evolution of M . The Eligible Interval of stream M has been labeled in grey context. The evolution of the total credit CR_H as well as the individual credit CR_{H_1}, CR_{H_2} and CR_{H_3} has been shown along with the frame transmission. The minimum credit of H and the maximum credit of M has been marked with dash-dotted lines. It has been shown that the total credit CR_H decreases at the slowest descend to reach the minimum, i.e., when one credit is decreasing and the rest two are always increasing. In this way, the maximum value of credit M is reached when the total credit of H reaches its minimum and the calculated worst-case relative delay is thus attained. The solid vertical line in red marks the timing when credit M reached the maximum value that we proved.

Chapter 6

Tightness of Our Approach

We have presented a relative worst-case response time analysis with interference consisting of multiple high and multiple low priorities in the previous chapter. We have proved that our worst-case bounds are safe. Another important factor of worst-case response time estimates is tightness. As we mentioned in Section 1.2, tightness describes how well a timing analysis method exploits the information provided to it; in other words, how close the estimated bounds are to what is theoretically possible within the given assumptions made in the traffic model. An analysis is considered tight if, within these assumptions, a scenario can be perceived that achieves the predicted bounds arbitrarily closely.

In this chapter, we study the conditions under which our relative worst-case response time analysis is tight. Firstly, we recapitulate our earlier result that it is always tight in case of a single high priority. Secondly, we use a number of examples to illustrate why tightness is not guaranteed in case of multiple high priorities. Thirdly, we recapitulate a quite elaborate scenario that reveals a sufficient (and conjectured necessary) condition on the high-priority classes under which our result is tight.

6.1 Tightness for a Single High-Priority Interference

To illustrate the worst-case scenario that underlies Theorem 5.4 for a single high priority, we repeat our construction in Cao et al. (2016a). Consider the left-most Eligible Interval in Figure 6.1. This figure shows how a maximum credit for class M , and hence the maximum delay for a frame from that class, can be achieved if we have *only a single* high-priority class H and one low-priority class L . The example, however, also works for multiple low-priority classes since only one frame of low-priority can transmit in

any case. The downward arrows indicate the arrival of frames; the solid rectangles indicate the transmission of the combined H, M , and L streams; the shaded rectangles indicate how the M stream would execute in a scenario without interference. The graphs below indicate the total credit level of both shapers and the individual credit levels of H and M during execution.

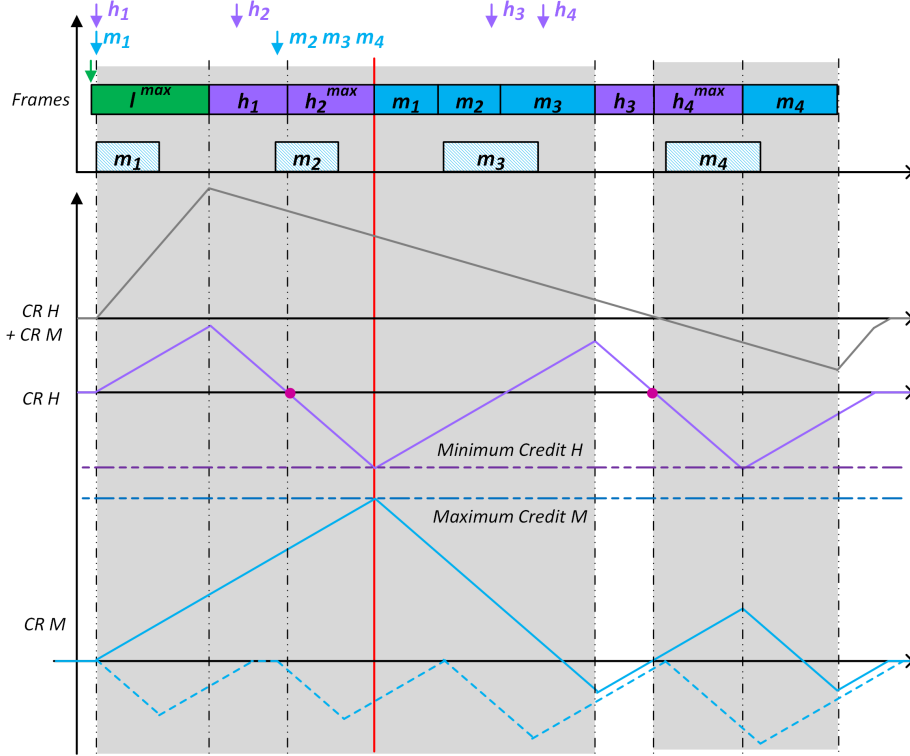


Figure 6.1: A stream of M interfered by a stream of H and L , and $\alpha_H^+ + \alpha_M^+ < BW$. The solid rectangles represent the transmission of streams H , M , and L , while the shaded ones represent the interference-free transmission of M . So do the solid and dashed lines representing the credit evolution. Frames of maximum transmission time have been labeled with superscript max, and the Eligible Interval has been labeled in grey context. The two dots on the curve of credit H emphasize the moment when credit reaches exactly zero and a frame of maximum size starts its transmission. The solid vertical line in red marks the timing when credit M reaches the maximum value that we determined.

In the worst-case scenario, the Eligible Interval starts with an L frame of maximum transmission time. Immediately after the start of L , the frame m_1 (our frame of interest) arrives together with an H frame. As a consequence, credit builds up both for M and H . Subsequently, frames from H

start transmitting, and new frames arrive until the credit built up by the L transmission is exactly depleted. This credit depletion takes a careful choice of frame sizes. Lastly, a maximal frame of H arrives before credit is entirely depleted to ensure that the credit of H reaches its minimal value (which in case of a single high-priority class equals $CR_H^{\min} = -\alpha_H^- \cdot C_H^{\max}$). Now, frame m_1 is ready to transmit, and at this point, the credit of M has reached its maximum value, and therefore frame m_1 has experienced its maximum delay compared to the uninterfered schedule. In the execution that follows in Figure 6.1, we see that the total credit keeps decreasing and that the maximum credit level of M is never obtained again. Only in case $\alpha_H^+ + \alpha_M^+ = BW$, a stream M may reach its maximum more than once in a single Eligible Interval because the total credit does not decrease.

To prove that the formula developed in Theorem 5.4 gives a tight bound on the worst-case relative delay when there is only a single high-priority class, it is enough to realize that the example illustrated in Figure 6.1 provides a generic construction of the worst case. For any given medium priority, the arrival of a low-priority frame and the immediately subsequent arrival of several high-priority frames of a well-chosen size can delay the transmission of any medium-priority frame by the amount specified in Corollary 5.4. This construction can be carried out for any frame in the medium-priority stream, as long as we have the freedom to assign a worst-case arrival time to a single low-priority frame and have the freedom to inject a burst of high-priority frames of the right size. Therefore, our approach is tight under the assumptions made in our system model, i.e., the estimate cannot be improved without further knowledge of the interference.

6.2 Examples Why Tightness Is Not Always Guaranteed for Multiple High-Priority Interference

To understand why tightness is not always guaranteed, recall the way to calculate the minimum credit of a set of N priority classes under Credit-Based Shaping, as presented in Theorem 5.1. To reach the minimum credit, one should ensure that the transmission in the last phase consisting of N frames, one of each priority level, are in the order determined by the iterative calculation of the minimum credit. Furthermore, recall that according to Theorem 5.4, this minimum credit should be reached at the slowest possible descent of the cumulative credit, while Theorem 5.3 shows that this slowest descent can be only attained if all credits are increasing at all times, except for the one that is transmitting.

We have specified a scenario in Table 6.1, which consists of four high-priority classes interfering with our medium-priority stream. To reach min-

imum cumulative credit for those four priority classes, one only needs to perform the execution of four maximum frames, one of each class, in the order H_2, H_3, H_1, H_4 , as illustrated in the left panel of Figure 6.2. Note that this construction is given in the proof of Theorem 5.1. To reach minimum credit at the slowest possible descent, however, one should make sure that the credit of each of the shapers is rising whenever it is not transmitting. Each transmission of the four maximum frames should start at 0 credit (otherwise, the minimum credit is not reached), which means that the credit of all shapers must be *recovering* before the start of transmission. This execution is illustrated in the right panel of Figure 6.2. Naturally, these recoveries can only be caused by earlier transmissions.

Table 6.1: An example of four priority levels ($H_1 > H_2 > H_3 > H_4$) in which reaching the minimum credit with the slowest decreasing rate is feasible in the last 4 transmissions.

	α^+ Mbps	α^- Mbps	C^{\max} μs	CR^{\min} bits
H_1	10	90	5	-450
H_2	10	90	6	-540
H_3	15	85	8	-680
H_4	10	90	4	-360

Now, to see why it is not always possible to reach the worst-case response time in the case of multiple high priorities, consider the slightly simpler scenario given in Table 6.2, where only two high-priority streams are interfering with our stream of interest. In this scenario, we have to transmit a maximal frame of H_2 followed by a maximal frame of H_1 in the last phase. As the left panel of Figure 6.3 indicates, reaching the minimum total credit is always possible without guaranteeing the slowest descent. However, to keep the slowest descent, the credit of H_1 needs to be recovering during the transmission of H_2 . The illustration on the right panel clearly shows that if H_1 has to recover exactly at the start of its final transmission, this recovery must have started at a lower value than the minimum credit of H_1 . Therefore, the minimum cumulative credit cannot be attained in the slowest possible way using this sequence of transmissions. Furthermore, we have not been able to find another sequence of transmissions that would have the desired result.

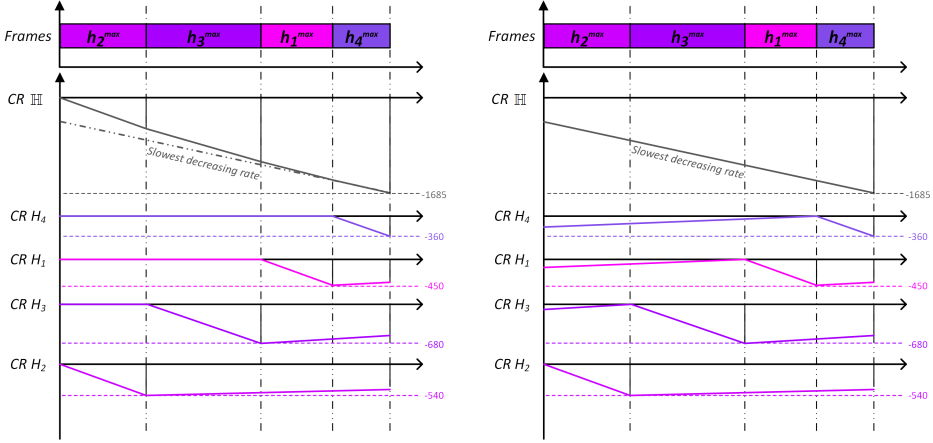


Figure 6.2: Example on how to reach minimum cumulative credit and reach it at the slowest possible descent. The last four transmissions are maximally-sized frames from H_2, H_3, H_1, H_4 , determined by the minimum total credit calculation, and the total minimum credit $CR_{\mathbb{H}}^{\min}$ is -1685 . Left panel: the total credit reaches the minimum but not at the slowest descent. Right panel: the total credit reaches the minimum at the slowest descent, i.e., during the transmission of one priority, all other credits are increasing. Note that the transmissions given in the left panel are the construction given in the proof of Theorem 5.1.

6.3 Tightness for Multiple High-Priority Interference

Now that we have shown tightness for a single interfering high priority, and given examples why tightness is not self-evident for multiple high-priority interference, let us study the latter in more depth. In case of a stream M interfered by a set of multiple high-priority classes \mathbb{H} and low-priority classes \mathbb{L} , our claims of tightness are conditional. Of course, still only one low-priority frame can interfere. However, as indicated before, if we try to create a scenario that achieves the worst-case, we need to aim for interference by the high-priority classes that at the same time a) reach the minimum total credit $CR_{\mathbb{H}}^{\min}$ and b) do so in the slowest feasible manner. In other words, we need a single scenario that serves as a witness for both Theorem 5.1 and 5.2. This is not always achievable, but it is possible to turn the combined construction of both witnesses into a sufficient condition for tightness.

In this dissertation, we just recapitulate the theorem in Cao et al. (2018b) and describe the core of the proof since we credit this proof to Pieter Cuijpers who came up with a feasible scenario to reach the worst-case and obtain the tightness. The detailed proof can be found in the appendix of the journal publication Cao et al. (2018b).

Table 6.2: An example of two priority levels ($H_1 > H_2$) in which reaching the minimum credit with the slowest decreasing rate is not feasible in the last 2 transmissions.

	α^+	α^-	C^{\max}	CR^{\min}
	Mbps	Mbps	μs	bits
H_1	20	80	1	-80
H_2	40	60	6	-360

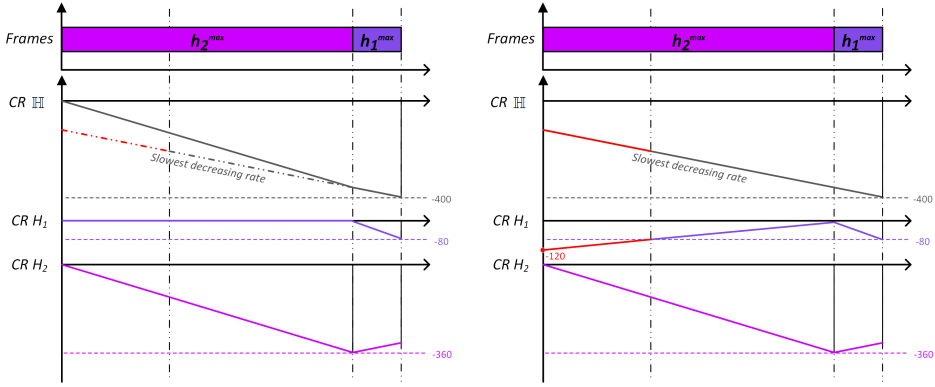


Figure 6.3: Example of a case where attaining minimum cumulative credit at the slowest descent is not possible. The last two transmissions are maximally-sized frames from H_2, H_1 , and the transmission order is determined by the minimum total credit calculation, and the total minimum credit $CR_{\mathbb{H}}^{\min}$ is -400 . Left panel: the total credit reaches the minimum but not at the slowest descent. Right panel: the total credit cannot reach the minimum and meanwhile decrease at the slowest descent. At the start of the first transmission, the credit of H_1 violates its minimum credit bound, resulting in an impossible scenario.

Theorem 6.1. *Given a stream of priority class $M \in \mathbb{P}$, we use $\mathbb{H} = \{X \in \mathbb{P} \mid X > M\}$ to denote all streams of high priorities and $\mathbb{L} = \{X \in \mathbb{P} \mid X < M\}$ to denote all streams of low priorities. Assume that M and \mathbb{H} are under Credit-Based Shaping and $\alpha_{\mathbb{H}}^+ + \alpha_M^+ \leq BW$. If there exists a sequence $H_1 \dots H_N \in \mathbb{H}$, with N the number of streams in \mathbb{H} , such that each priority class only occurs once in the sequence (but not necessarily in order of priority), and if for all $1 \leq n \leq N$:*

$$CR_{\bigcup_{1 \leq i \leq n} H_i}^{\min} = -\alpha_{\bigcup_{1 \leq i \leq n} H_n}^- \cdot C_{H_n}^{\max} + CR_{\bigcup_{1 \leq i < n} H_i}^{\min}, \quad (6.1)$$

and furthermore

$$C_{H_N}^{\max} \geq \frac{\alpha_{H_N}^+}{\alpha_{H_N}^-} \sum_{1 \leq j < N} C_{H_j}^{\max}, \quad (6.2)$$

then the bound on the worst-case relative delay of Theorem 5.4 is tight.

As a brief sketch of the proof, a generic construction u^ϵ of one feasible scenario is presented, which maintains the slowest possible decrease of credit at all times to reach the minimum credit, except for an arbitrarily small time during which an arbitrarily small drop in credit may take place due to resets.

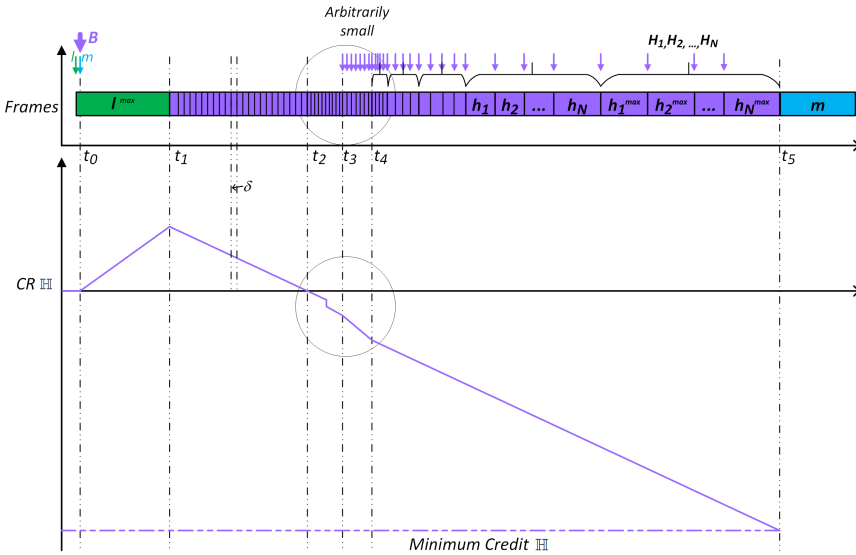


Figure 6.4: Construction of an arbitrary approximation of the worst-case response time in five phases. In the first phase, the total credit of high priorities builds up during the transmission of a maximum-sized low-priority frame. In the second phase, the built-up credit is slowly depleted until zero, and in the fifth and final phase, the credit is slowly depleted until a minimum credit is reached. The third and fourth phase allow for an arbitrarily small drop in credit during an arbitrarily small amount of time, constructed in such a way that a valid schedule between second and fifth phase is created.

Admittedly, the scenario is quite complicated. It consists of five phases, illustrated in Figure 6.4. The first phase, like in the single high-priority case, consists of a transmission of a low-priority frame of the maximum size, while all other shapers receive pending load just after the transmission of the low-priority frame has started. This is indicated in Figure 6.4 by a burst B .

In the second phase, the cumulative credit that has increased during the first phase is depleted using the burst B , which is a large burst of high-priority frames of arbitrarily small size $\delta > 0$. As a consequence, the cumulative credit reaches 0, while all high-priority shapers have pending load at all times, and at least one of high priorities is transmitting at all times. This means, using Theorem 5.3, that credit drops at its slowest possible rate during this second phase.

In the third and fourth phase, which only take an arbitrarily short amount of time, the cumulative credit may drop by a small amount because some (but not all) shapers experience a credit reset. After this reset, a number of arbitrarily small transmissions take place to put the credit of all high-priority shapers at the right (negative) amount to start with the fifth phase.

Finally, in the fifth phase, credit of all high-priority shapers remains negative at all times, except for the one that needs to transmit, which has credit 0 exactly when needed. We use transmissions in order of the sequence $H_1 \dots H_N \in \mathbb{H}$, all with the maximum frame-sizes. If furthermore the sequence satisfies the (strictly stronger) conditions given in the theorem then the last iteration of transmissions $H_1 \dots H_N$ consist of maximal frame sizes, resulting in the minimum credit. Therefore, the execution u^ϵ is a witness of an arbitrary approximation of the worst-case response time for a frame $m \in M$.

Conjecture 6.1. *We conjecture that the conditions given in Theorem 6.1 are in fact necessary and sufficient for tightness.*

The conjecture is based on the example discussed in the previous section with a focus on the final phase. It is certainly necessary to approximate minimum credit in order to reach the worst-case response time, and any sequence of transmissions approximating minimum credit has to use frames of maximal sizes. (Assuming otherwise leads to a contradiction, since a larger frame-size immediately leads to lower credit). The conditions in Theorem 6.1 exactly describe a possible last sequence of transmissions needed to actually reach minimum credit. We conjecture that this is, in fact, the only possible sequence that leads to minimum credit, but a formal proof of this is left for future research.

Finally, note that the complexity of checking the condition for tightness just involves one inequality check once the order of the sequence to reach the minimum credit is determined by Theorem 5.1. The complexity of the minimum credit analysis is factorial in the number of high-priority streams as explained in Section 5.1.

6.4 Conclusion

After presenting a technique to perform worst-case response time analysis, independent of knowledge of the interfering traffic (except those assumptions enforced by the Ethernet TSN standard), we further proved sufficient (and conjectured necessary) conditions under which our analysis yields a tight bound. Notably, the complexity of our worst-case relative delay analysis mainly depends on the complexity of the minimum credit calculation, which is factorial in the number of high-priority classes (see Theorem 5.1), and the complexity of checking the sufficient conditions just involves an additional inequality check. Unlike the more traditional busy period analysis, that was proposed in, amongst others, Axer et al. (2014) and Bordoloi et al. (2014), our method does not rely on recursions of which the depth is dependent on the detailed traffic models in the system.

Chapter 7

Comparison with Earlier Works

In this chapter, we perform a comparison of our relative worst-case time analysis with the busy period analyses presented in Axer et al. (2014) and Bordoloi et al. (2014). For this, we focus on scenarios in which there is only a single high priority. In this case, our approach is tight, assuming there is no additional information about the traffic model regarding the interference. However, in the works of Axer et al. (2014) and Bordoloi et al. (2014) interference is assumed to be periodic, or at least have a minimum inter-arrival time. Because of these additional assumptions, the approach of Axer et al. (2014) and Bordoloi et al. (2014) may theoretically provide improved bounds compared to the approach presented in this dissertation. We investigate whether that is the case, i.e., whether the additional assumptions help to improve the worst-case response time estimates further.

To compare the works of Axer et al. (2014) and Bordoloi et al. (2014) with ours, we first use the same periodic traffic model to calculate the worst-case response time in an uninterfered schedule and then add the relative worst-case response time on top of that without restricting the interference to any particular model. It turns out that the approaches of Axer et al. (2014) and Bordoloi et al. (2014) do not help to improve our estimate any further. On the contrary, we identify that some pessimism present in the approaches of Axer et al. (2014) and Bordoloi et al. (2014) can be remedied by the use of relative worst-case time analysis.

When trying to generalize the comparison further, it was discovered that performing a fair comparison in the case of multiple high priorities was difficult. In both Axer et al. (2014) and Bordoloi et al. (2014), there

is the implicit suggestion that multiple high priorities may be addressed by summing up the interference from all high priorities. However, no clear calculation and results were provided for this case, and the authors only presented examples in their papers that addressed a single high priority. Extending their works turned out not to be trivial, and we therefore did not have any clear results to compare against.

7.1 Worst-Case Response Time Analysis in Uninterfered Schedules

The worst-case time analysis given in Theorem 5.4, which is the main theorem of Chapter 5, provides a relative result. In case of a single high priority, it states that the worst-case response time of any class M frame is tightly bounded by its worst-case response time in an uninterfered schedule plus a constant term determined by the maximum frame size of high- and low-priority traffic and the reservation of the high-priority shaper (see Corollary 5.4). To compute the worst-case response time of a frame, just knowing the relative delay compared to an uninterfered schedule is not sufficient. We also need to calculate the worst-case response time of that uninterfered schedule. Thus, it is necessary to make some assumptions on the class M traffic, which causes intra-class interference. Up to this point, such assumptions do not play a part in our overall system model. For the main contribution of this dissertation, it is not essential how the uninterfered schedule is analyzed, but when comparing our work to that of others, it is important that we have a way to do so.

The worst-case response time analysis of an uninterfered schedule with a single priority is much simpler compared to that of an interfered schedule with multiple priorities. The credit decreases to negative when there is a transmission and recovers to zero before another transmission takes place. We can consider it as a simple FIFO schedule (without Credit-Based Shaping) with an inflated transmission time, which takes the recovery times of Credit-Based Shaping as a part of the transmission. The only difference is that we should not consider the recovery time of the last frame.

In this section, we start with the worst-case response time analysis in FIFO executions without any shaping involved. In order to make the conclusion derived based on FIFO schedules more easily extendable to uninterfered schedules under Credit-Based Shaping, the notion of worst-case waiting times is introduced. The *worst-case waiting time* is the maximum duration that it takes for a frame to arrive at the queue and start its transmission; it is basically the worst-case response time without the transmission time of the frame itself. We introduce this notion because we can simply

scale the worst-case waiting time from a FIFO to an uninterfered schedule and then we can easily add a maximum transmission time to get the worst-case response time in an uninterfered schedule.

Theorem 7.1. *Given a single-priority stream X composed of finite periodic sources and each source τ_i has the maximum frame size C_i and period T_i , and all sources are undergoing FIFO transmission. If the total utilisation does not exceed the total bandwidth, i.e., $\sum_{\tau_i \in X} \frac{C_i}{T_i} \leq 1$, the worst-case waiting time for each single source τ_i in all FIFO executions \mathbb{V}_0 is determined by:*

$$WW^{\mathbb{V}_0}(\tau_i) = \sum_{\tau_j \in X, j \neq i} C_j. \quad (7.1)$$

Proof. We determine the worst-case waiting time of all possible FIFO executions \mathbb{V}_0 by computing an upper bound of the preceding load of the source at any time. In this proof, it is important to note that we only deal with FIFO executions without any shaping involved. The *preceding load* of each source τ_i at time $t \geq 0$ is denoted as $P_i(t)$, which is the total pending load $P(t)$ excluding the transmission time of the last frame in source τ_i . While the upper bound of the total pending load determines the worst-case response time of source τ_i , the upper bound of the preceding load $P_i(t)$ determines the maximum waiting time of the last frame in source τ_i .

Firstly, we denote the cumulative load as $R(t)$ and the cumulative processed load as $R'(t)$. The difference of these two loads is the total pending load, i.e., $P(t) = R(t) - R'(t)$. Furthermore, we denote the phase, the arrival time of the first instance, for each periodic source as $0 \leq \phi_i < T_i$ and we assume that the n^{th} instance of transmission of source i has size $0 \leq c_i(n) \leq C_i$. Obviously, the cumulative load $R(t)$ is then determined by

$$R(t) = \sum_{\tau_i \in X} \sum_{0 < n \leq \frac{t - \phi_i}{T_i} + 1} c_i(n), \quad (7.2)$$

and the preceding load any $t \geq \phi_i$ is

$$P_i(t) = P(t) - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor). \quad (7.3)$$

Secondly, we are going to prove the cumulative processed load $R'(t)$ to be the infimum of $R(t_1) + t - t_1$ for any $0 \leq t_1 \leq t$, following the proof in Cuijpers and Bril (2006). For a single non-idling server, it is obvious that the process load at time t can never be more than the processed load

at a time $t_1 \leq t$ plus the load that can be processed between t_1 and t . If the buffer is always non-empty, this load reaches its maximum $t - t_1$; If the buffer is empty at some point, this load will be less. Furthermore, the cumulative processed load can never be more than the cumulative load, i.e., $R'(t) \leq R(t)$. Hence we have:

$$\begin{aligned} R'(t) &\leq \max\{(R'(t_1) + t - t_1), R(t)\} \\ &\leq \max\{(R(t_1) + t - t_1), R(t)\}. \end{aligned} \quad (7.4)$$

Since this holds for all $t_1 \leq t$, it also holds for the infimum over t_1 :

$$R'(t) \leq \max\left\{\inf_{0 \leq t_1 \leq t} \{R(t_1) + t - t_1\}, R(t)\right\}. \quad (7.5)$$

Using the special case where $t_1 = t$, we can simplify this to:

$$R'(t) \leq \inf_{0 \leq t_1 \leq t} \{R(t_1) + t - t_1\}. \quad (7.6)$$

For a non-idling server, we can derive a lower bound on $R'(t)$ as well. We define t_0 as the latest point before t at which the buffer is empty.

$$t_0 \stackrel{\text{def}}{=} \sup\{t' | R(t') = R'(t'), 0 \leq t' \leq t\} \quad (7.7)$$

Between t_0 and t , the buffer is always non-empty, which means that the server always processes the load, so using $R(t_0) = R'(t_0)$ we have the equality:

$$\begin{aligned} R'(t) &= \max\{(R'(t_0) + t - t_0), R(t)\} \\ &= \max\{(R(t_0) + t - t_0), R(t)\}, \\ &\geq \max\left\{\inf_{0 \leq t_1 \leq t} \{R(t_1) + t - t_1\}, R(t)\right\}, \\ &= \inf_{0 \leq t_1 \leq t} \{R(t_1) + t - t_1\}, \end{aligned} \quad (7.8)$$

Using Equation (7.6) and Equation (7.8), we conclude that $R'(t)$ is exactly determined by

$$R'(t) = \inf_{0 \leq t_1 \leq t} \{R(t_1) + t - t_1\}. \quad (7.9)$$

Thirdly, we determine the upper bound on the preceding load. With the assumption $\sum_{\tau_i \in X} \frac{C_i}{T_i} \leq 1$, we derive the pending load at any time $t \geq 0$ as:

$$\begin{aligned}
& P_i(t) \\
&= P(t) - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor) \\
&= R(t) - R'(t) - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor) \\
&= R(t) - \inf_{0 \leq t_1 \leq t} \{R(t_1) + t - t_1\} - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor) \\
&= \sup_{0 \leq t_1 \leq t} \{R(t) - R(t_1) - t + t_1 - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor)\} \\
&= \sup_{0 \leq t_1 \leq t} \{ \sum_{\tau_j \in X} \sum_{0 < n \leq \frac{t - \phi_j}{T_j} + 1} c_j(n) - \sum_{\tau_j \in X} \sum_{0 < n \leq \frac{t_1 - \phi_j}{T_j} + 1} c_j(n) - t + t_1 - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor) \} \\
&= \sup_{0 \leq t_1 \leq t} \{ \sum_{\tau_j \in X} \sum_{\max\{0, \frac{t_1 - \phi_j}{T_j} + 1\} < n \leq \frac{t - \phi_j}{T_j} + 1} c_j(n) - c_i(\lfloor \frac{t - \phi_i}{T_i} + 1 \rfloor) - t + t_1 \} \\
&= \sup_{0 \leq t_1 \leq t} \{ \sum_{\tau_j \in X, j \neq i} \sum_{\max\{0, \frac{t_1 - \phi_j}{T_j} + 1\} < n \leq \frac{t - \phi_j}{T_j} + 1} c_j(n) + \sum_{\max\{0, \frac{t_1 - \phi_i}{T_i} + 1\} < n \leq \frac{t - \phi_i}{T_i}} c_i(n) - t + t_1 \} \\
&\leq \sup_{0 \leq t_1 \leq t} \{ \sum_{\tau_j \in X, j \neq i} \sum_{\max\{0, \frac{t_1 - \phi_j}{T_j} + 1\} < n \leq \frac{t - \phi_j}{T_j} + 1} C_j + \sum_{\max\{0, \frac{t_1 - \phi_i}{T_i} + 1\} < n \leq \frac{t - \phi_i}{T_i}} C_i - t + t_1 \} \\
&\leq \sup_{0 \leq t_1 \leq t} \{ \sum_{\tau_j \in X, j \neq i} \lfloor \frac{t - t_1}{T_j} + 1 \rfloor \cdot C_j + \lfloor \frac{t - t_1}{T_i} \rfloor \cdot C_i - t + t_1 \} \\
&\leq \sup_{0 \leq t_1 \leq t} \{ \sum_{\tau_j \in X, j \neq i} (\frac{t - t_1}{T_j} + 1) \cdot C_j + (\frac{t - t_1}{T_i}) \cdot C_i - t + t_1 \} \\
&= \sup_{0 \leq t_1 \leq t} \left\{ \left(\sum_{\tau_j \in X} (\frac{t - t_1}{T_i}) \cdot C_i \right) - t + t_1 + \sum_{\tau_j \in X, j \neq i} C_j \right\} \\
&= \sup_{0 \leq t_1 \leq t} \left\{ \left(\sum_{\tau_i \in X} \frac{C_i}{T_i} - 1 \right) \cdot (t - t_1) + \sum_{\tau_j \in X, j \neq i} C_j \right\} \\
&\leq \sum_{\tau_j \in X, j \neq i} C_j.
\end{aligned} \tag{7.10}$$

The maximum preceding load is bounded by the sum of maximum frame sizes from all other sources except the source of interest. This illustrates one possible worst-case scenario when all other sources simultaneously release one frame of the maximum size, and the source of interest releases one frame at the same time and starts its transmission at the last, resulting in the worst-case waiting time of that source.

□

Corollary 7.1. *Given a single-priority stream X composed of a finite number of periodic sources and each source τ_i has the maximum frame size C_i and period T_i , and all sources are undergoing FIFO transmission. If the total utilisation does not exceed the total bandwidth, i.e., $\sum_{\tau_i \in X} \frac{C_i}{T_i} \leq 1$, the worst-case response time for each single source in all FIFO executions \mathbb{V}_0 is determined by:*

$$WR^{\mathbb{V}_0}(\tau_i) = \sum_{\tau_j \in X} C_j. \quad (7.11)$$

Next, observe that an uninterfered schedule of a Credit-Based Shaper behaves in much the same way as a FIFO queue, except that each transmission is followed by a recovery period proportional to the size of the transmission. As a result, we can consider the recovery time as part of the load. The only difference is that we should not include the recovery time of the last frame when computing the worst-case response time.

Theorem 7.2. *Given a single-priority stream X composed of a finite number of periodic sources and each source τ_i has the maximum frame size C_i , and period T_i , and all sources are undergoing a Credit-Based Shaper with a reservation α_X^+ . If the total utilisation does not exceed the reserved bandwidth, i.e., $\sum_{\tau_i \in X} \frac{C_i}{T_i} \leq \frac{\alpha_X^+}{BW}$, then the worst-case response time for each single source in all uninterfered executions \mathbb{U}_0 is determined by:*

$$WR^{\mathbb{U}_0}(\tau_i) = \left(\sum_{\tau_j \in X, j \neq i} C_j \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right) \right) + C_i. \quad (7.12)$$

Proof. In Section 4.2, it is explained how transmission works under Credit-Based Shaping in an uninterfered schedule, see Property 4.1. In such a schedule, the credit decreases at a rate of α_X^- during the transmission and recovers at a rate of α_X^+ to zero before another frame can be transmitted. The recovery time is the product of the frame transmission time and the ratio α_X^-/α_X^+ , see Figure 4.1. As a result, we can take transmissions under Credit-Based Shaping in an uninterfered schedule as FIFO transmissions when we consider the recovery time as part of the transmission load. In the calculation of the worst-case waiting time of Credit-Based Shaping in an uninterfered schedule this translates to taking the worst-case waiting time of FIFO transmission derived in Theorem 7.1 and multiplying by a factor $1 + \frac{\alpha_X^-}{\alpha_X^+}$:

$$WW^{\mathbb{U}_0}(\tau_i) = \sum_{\tau_j \in X, j \neq i} C_j \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right). \quad (7.13)$$

Then, adding a maximum frame size C_i as the upper bound of the transmission time of the last frame of τ_i gives us the worst-case response time:

$$WR^{\mathbb{U}_0}(\tau_i) = \left(\sum_{\tau_j \in X, j \neq i} C_j \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right) \right) + C_i. \quad (7.14)$$

Note that the utilization condition is necessary and sufficient to guarantee that the uninterfered schedule has finite worst-case response times. In Theorem 7.1, it is $\sum_{\tau_i \in X} \frac{C_i}{T_i} \leq 1$. Here, we multiply by a factor of $1 + \frac{\alpha_X^-}{\alpha_X^+}$ on the left side of the condition:

$$\sum_{\tau_i \in X} \frac{C_i}{T_i} \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right) \leq 1. \quad (7.15)$$

Observing that $1 + \frac{\alpha_X^-}{\alpha_X^+} = \frac{BW}{\alpha_X^+}$, we derive:

$$\sum_{\tau_i \in X} \frac{C_i}{T_i} \leq \frac{\alpha_X^+}{BW} \quad (7.16)$$

as a utilization condition. □

Then, adding the interference according to Theorem 5.4 gives us:

$$\begin{aligned} & WR^{\mathbb{U}}(\tau_i) \\ &= WR^{\mathbb{U}_0}(\tau_i) + C_L^{\max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{\max} \\ &= \left(\sum_{\tau_j \in X, j \neq i} C_j \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right) \right) + C_i + C_L^{\max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{\max}. \end{aligned} \quad (7.17)$$

7.2 Illustrative Example: Single High-Priority Interference

As an arbitrarily chosen illustrative example, let us consider three sources of class M traffic: τ_1, τ_2, τ_3 . In line with the assumptions in Bordoloi et al. (2014), we assume that each source has periodic behavior characterized by the inter-arrival time T_i and a maximum frame-size C_i . Note that we only

need to assume this for the class M traffic, not for the high- and low-priority sources of interference. For the latter, we only assume maximum frame sizes C_L^{\max} and C_H^{\max} , and the reservation of high priority α_H^+ . This is summarized in Table 7.1, where we assign values to those parameters. Finally, we assume a bandwidth $BW = 100$ Mbps in this table and choose the respective reservations of the shapers to be 40% of this, so: $\alpha_H^+ = \alpha_M^+ = 40$ Mbps and $\alpha_H^- = \alpha_M^- = 60$ Mbps.

Under the condition that the utilization of the sources is less than the reservation, i.e., $\sum_{\tau_i \in M} \frac{C_i}{T_i} \leq \frac{\alpha_M^+}{BW}$, we find that the worst-case response time in the uninterfered schedule is finite. We can determine the worst-case response time for any frame from source τ_i using Equation (7.17), which returns the values in the right part of Table 7.1.

Table 7.1: Example scenario, calculating worst-case response times for class M sources τ_1, τ_2 and τ_3 using the analysis presented in this dissertation, given a total bandwidth of 100 Mbps. The parameters on priority classes are in grey.

	P	α^+ Mbps	C μ s	C_i μ s	T_i μ s	WR μ s
	H	40	1	-		-
τ_1	M	40	3	1	25	17.83
τ_2				3	30	14.83
τ_3				2	20	16.33
	L	-	2	-		-

7.3 Applying Existing Methods to the Same Example

Now, in order to compare our approach with that of Axer et al. (2014) and Bordoloi et al. (2014), we must first observe that in those approaches more knowledge regarding the interfering traffic is assumed than in ours. In particular, in Bordoloi et al. (2014) it is assumed that also the high-priority traffic is characterized as a set of periodic sources. Moreover, a deadline associated with each high-priority task is used in the calculations of Bordoloi et al. (2014) that compensates for the fact that some high-priority traffic transmits during the recovery time. In Axer et al. (2014), the Compositional Performance Analysis approach is used, meaning that high- and medium-priority traffic are characterized by arrival curves instead of as periodic sources. An arrival curve specifies the maximal and minimal amount of traffic in an arbitrary interval of a certain size. In this comparison, we follow the assumptions of Bordoloi et al. (2014), because it is a strictly

stronger assumption than that of Axer et al. (2014). Any periodic source can easily be represented as a (periodic) arrival curve.

Potentially, the addition of information on the interference means that the worst-case performance bounds can be improved over ours, even though our approach is tight for our system model. Further on, we present examples when this is indeed the case, but also examples where both Axer et al. (2014) and Bordoloi et al. (2014) are pessimistic.

It is outside the scope of this dissertation to repeat the algorithms presented by Axer et al. (2014) and Bordoloi et al. (2014), but for completeness, we must mention a small adjustment that is needed before we can compare the work of Bordoloi et al. (2014) to ours and that of Axer et al. (2014). We observe that the definition of worst-case response time in Bordoloi et al. (2014) includes the recovery time after the transmission of interest, while in our approach and that of Axer et al. (2014), the response time ends immediately after the transmission. In order to provide a fair comparison, we have (straightforwardly) adapted Equations (20) and (21) in Bordoloi et al. (2014) to reflect this. Also, in the second improvement of Bordoloi et al. (2014), the calculation of medium-priority recovery time introduces unnecessary pessimism, and we adapt it according to Axer et al. (2014). Furthermore, in Axer et al. (2014), Equation (18) contains a term I_{HPB} , used to bound the time during which credit of a shaper can build up before a burst. The exact way in which the authors calculate I_{HPB} , however, remains unclear from the paper. Therefore, we decided to use our own estimate of the maximum attainable credit instead.

Table 7.2 contains the results of applying the methods from Axer et al. (2014) and Bordoloi et al. (2014) to our illustrating example. We refer to the online appendix Cao (2016) for the Matlab codes that are used to create this table, and the graphs further on. We have refined the information on high-priority tasks, by distinguishing two sources τ_4 and τ_5 , and we have added deadlines for both tasks. It can be noted immediately, that despite this added information, the results for this particular example are more pessimistic when compared to those obtained using our approach, as shown in Table 7.1.

This is surprising, because the worst-case scenario that goes with this example, displayed in Figure 7.1, shows that the burst of high-priority behavior is too small to fully deplete the credit that has been built up. The maximum credit for M is not reached and the worst-case response time predicted by our approach is thus not met. That is to say, in this example, our approach is already pessimistic since there is not enough flexibility to generate the necessary worst-case interference. Given that comparative studies

have provided more pessimistic results, we set out to investigate the sources of the pessimism in Axer et al. (2014) and Bordoloi et al. (2014) that cause the overestimation of the worst-case response time.

Table 7.2: Example scenario, calculating worst-case response times for class M sources τ_1, τ_2 and τ_3 using the approaches in Axer et al. (2014) and in Bordoloi et al. (2014), given a total bandwidth of 100 Mbps. The parameters on priority classes are in grey.

	P	α^+ Mbps	C μs	C_i μs	T_i μs	D_i μs	WR (Axer) μs	WR (Bordoloi) (μs)
τ_4	H	40	1	1	10	10	-	-
τ_5				1	5	10	-	-
τ_1	M	40	3	1	25	-	23.5	18.5
τ_2				3	30	-	17.5	16.5
τ_3				2	20	-	20	19
-	L	-	2	-			-	-

7.4 Exploring Pessimism

The busy period analyses of Axer et al. (2014) and Bordoloi et al. (2014) start out from the same basic analysis. Both papers initially identify a model in which four sources of interference are identified: high-priority interference, low-priority interference, medium-priority interference, and interference due to shaping. Incidentally, this basic analysis in the two papers coincides for periodic sources, and for reference we have added the results of this analysis (under the name ‘basic busy period analysis’) in the figures that occur in this section.

Both Axer et al. (2014) and Bordoloi et al. (2014) start by observing that the traditional calculation of high-priority interference, adding up all arrivals during a busy period, disregards the fact that the high priority is shaped. This source of pessimism is taken care of by adding a burst-rate model for the shaper.

In Figure 7.2, we illustrate the influence of this improvement, by considering the same shaper reservations and maximum frame sizes as in the grey parts in Table 7.1. We determine the worst-case response time of a single medium-priority source with $C = 3$, and $T = 30$, interfered by n_H identical high-priority sources with $C_i = 1$, $T_i = D_i = 5 \cdot n_H$, as well as a low-priority stream. Note, that we keep the total utilization of H at 20 Mbps (below its reservation α_H^+) as n_H ranges from 1 to 20.

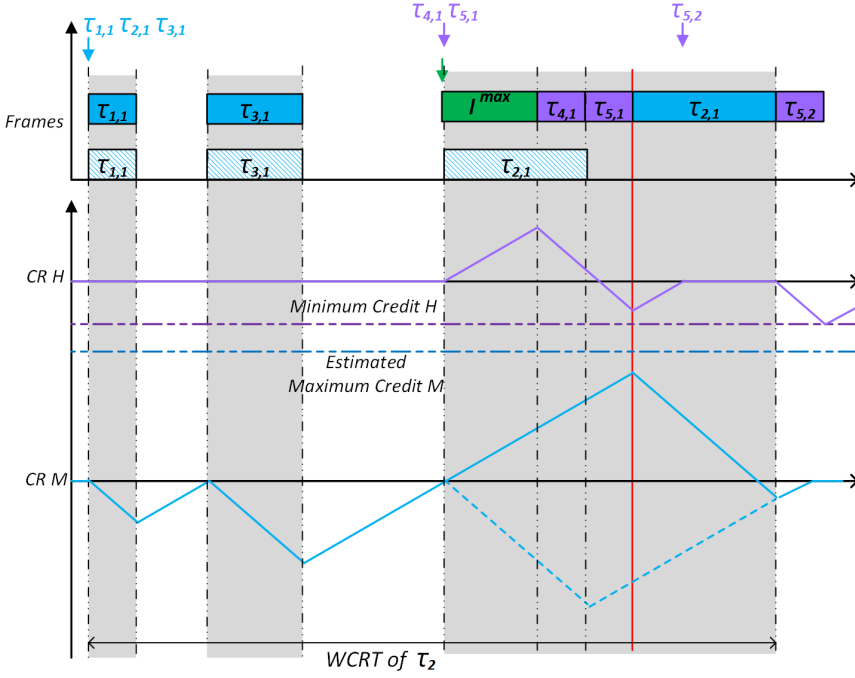


Figure 7.1: Worst-case scenario for τ_2 when adding knowledge regarding periodic interference (see Table 7.2). A maximum low-priority frame is released just before transmission of τ_2 is enabled, with subsequent arrivals of frames from high-priority τ_4 and τ_5 . Note how these two frames are insufficient to deplete the credit of H to its theoretical minimum. Therefore, a maximum built-up of credit of M is prevented, causing the transmission of τ_2 to start earlier than estimated.

The result of varying the number of high-priority sources is that the possible burst size of high-priority traffic increases. The figure clearly shows how, for low values of n_H , the burst size is smaller than the limits set by the Credit-Based Shaper. For those low values, the predictions of Axer et al. (2014) and Bordoloi et al. (2014) turn out to be better than ours, because we do not consider the load generated by high-priority sources. For higher values of n_H , the high-priority shaping becomes dominant, and those of Axer et al. (2014) and Bordoloi et al. (2014) eventually stabilize at a fixed value, while the basic busy period analysis keeps growing because it does not take high-priority shaping into account at all. Our approach always remains at the same value, which is tight and independent of the pattern of high-priority traffic.

The reason why Bordoloi et al. (2014) performs better than Axer et al.

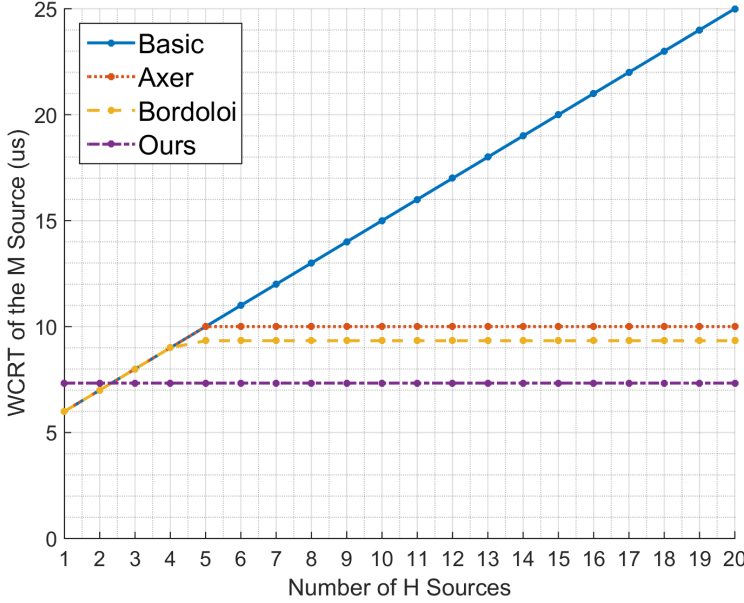


Figure 7.2: Worst-case response time of a medium-priority source given interference by n_H identical high-priority sources and a low-priority stream.

(2014) in Figure 7.2, is because of a second cause of pessimism, which was not addressed in Axer et al. (2014). This cause of pessimism was already briefly discussed in the introduction, when we mentioned that it is complicated (to say the least) to apply busy period analysis to idling servers. The four types of interference considered in the basic busy period analysis turn out not to be independent. If part of the high-priority traffic transmits during a recovery period of the medium-priority shaper, then this high-priority traffic should actually not be counted as interference. This is because it does not contribute to the busy period. The authors of Bordoloi et al. (2014) realised this and have made an attempt to take this effect into account.

This is illustrated in Figure 7.3, in which we again consider the same shaper reservations and maximum frame sizes as in Table 7.1. As before, we determine the worst-case response time of a single medium-priority source with $C = 3$, and $T = 30$, but this time the interference consists of a low-priority stream, a single high-priority source with $C = 1$, and $T = D = 5$, and n_M medium-priority sources with $C_i = 1$, $T_i = D_i = 5 \cdot n_M$.

By increasing the number of medium-priority tasks, we increase the recovery time as well. Furthermore, by choosing a single high-priority task with relatively small frame size, we ensure that the maximally built-up

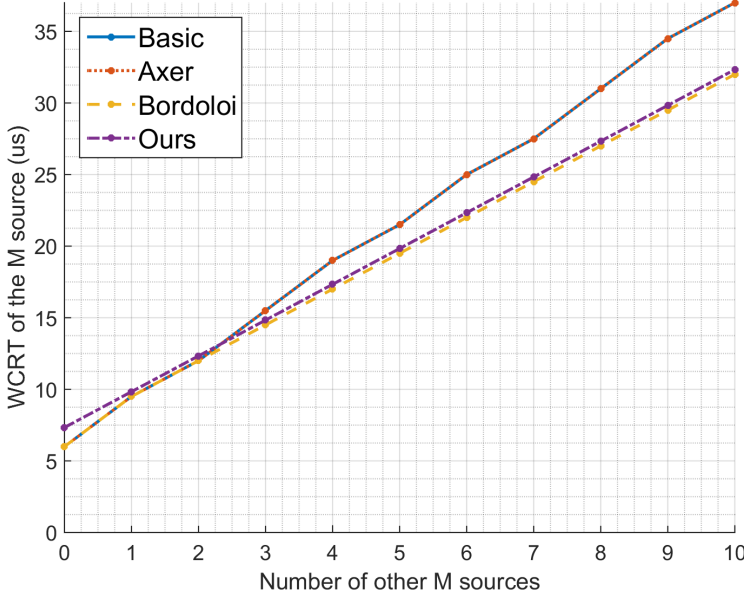


Figure 7.3: Worst-case response time of a medium-priority source given interference by one high-priority source, n_M medium-priority sources, and a low-priority stream.

credit cannot be fully depleted. As a consequence, the approach of Bordoloi et al. (2014) is consistently better than ours in Figure 7.3, while the approach of Axer et al. (2014) coincides with the basic busy period analysis and becomes more and more pessimistic as n_M grows.

Finally, if we adapt the scenario in which there are still n_M interfering medium-priority sources, and consider $n_H = 15$ identical high-priority sources instead of just 1, we see a completely different picture, shown in Figure 7.4. Now, our approach is tight again, because there is sufficient high-priority traffic to deplete the credit to reach its minimum. Furthermore, if we range n_M from 1 to 100, we see that the approaches of Axer et al. (2014) and Bordoloi et al. (2014) give better estimates than the basic analysis for low values ($n_M \leq 6$), since the shaping of high-priority traffic is taken into account. For high values, the approach of Axer et al. (2014) coincides with the basic analysis, and shows a ‘staircase’ behavior in its pessimism which indicates that the arrival of high-priority interference is the limiting factor in the analysis. Interestingly, the analysis from Bordoloi et al. (2014) suffers from the same over approximation until $n_M = 47$, and only then merits from the implemented improvements.

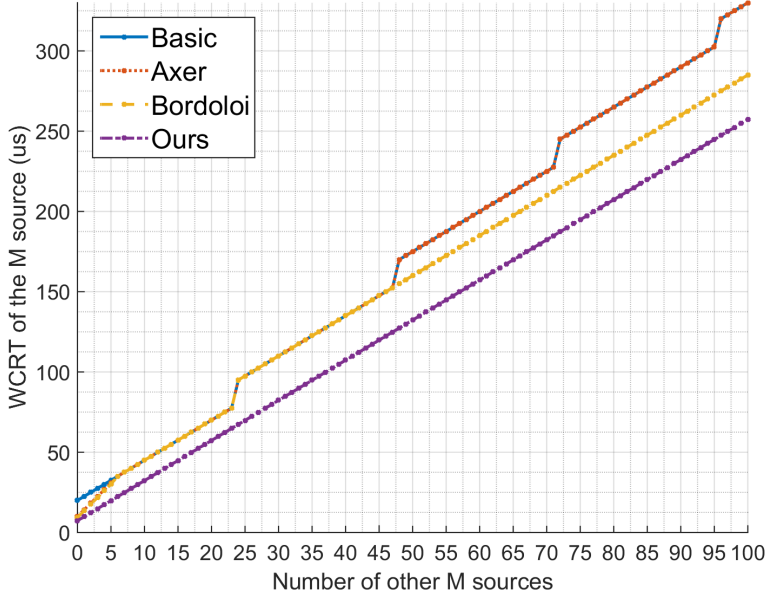


Figure 7.4: Worst-case response time of a medium-priority source given interference by 15 high-priority sources, n_M medium-priority sources, and a low-priority stream.

The complexity of the approach set out in Bordoloi et al. (2014) makes it difficult to find a satisfactory explanation for the pessimism that still is present. Instead, it may be more promising to consider whether our own approach can be improved in cases where the burstiness of high-priority traffic is insufficient to generate the worst-case behavior predicted by our approach.

7.5 Adding Information

From the comparison so far, we conjecture that the only pessimism that remains in our approach occurs when the high-priority traffic cannot produce a sufficiently large burst. Using an iterative approach similar to busy period analysis, we can calculate the maximum time *Burst* during which there can be a continuous transmission of high-priority traffic. Considering the scenario in Figure 7.1, we claim that, given knowledge of this burst, our worst-case response time analysis can be adapted to:

$$\begin{aligned}
& WR^{\mathbb{U}}(\tau_i) \\
& = WR^{\mathbb{U}_0}(\tau_i) + \min \left(C_L^{\max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-} \right) + C_H^{\max}, C_L^{\max} + Burst \right). \quad (7.18)
\end{aligned}$$

Obviously, the burstiness only makes a difference if *Burst* is smaller than the time needed to deplete the built-up credit plus a single maximum high-priority frame transmission.

Naturally, one can go even further, and consider the fact that the last transmission must be a frame of maximum size, and that the transmissions preceding this must exactly fit the depletion time of the maximum credit. If those transmissions do not fit exactly (for instance, because frames have a fixed size) the worst-case can also not be attained. But such considerations only complicate the analysis, and are expected to lead to relatively small improvements compared to the results already achieved by our Eligible Interval analysis.

7.6 Conclusion

In this chapter, we proved a methodology how to compute the worst-case response times of multiple periodic sources of one single priority, with which we carried out a comparison of our relative worst-case response time analysis with the state-of-the-art works (Axa et al. (2014) and Bordoloi et al. (2014)).

The outcome of this comparison (regarding only one single high priority) suggests that our approach is superior in situations where the burstiness of high-priority traffic is large enough. In particular, our approach is superior when the burst is larger than the depletion time of the maximum credit built up by the low-priority traffic plus a single maximum high-priority frame transmission. Furthermore, given the input models assumed in Axa et al. (2014) or Bordoloi et al. (2014), we conjecture it is fairly easy to estimate the maximum burst size and slightly improve our analysis. This would give an equivalent result to the state at the art. However, such an improvement requires additional assumptions on the sources of interference. It greatly depends on the type of system under study whether these assumptions can be made, in particular that may not be the case when a network contains a large number of third party components. An independent analysis is greatly desirable in such cases.

Chapter 8

Relative Best-Case Response Time Analysis

8.1 Introduction

In the earlier chapters of this dissertation, we have spent a significant amount of time formally proving an upper bound of the relative worst-case response times. Next to the worst-case analysis, best-case response time analysis becomes important whenever timing constraints impose lower bounds on response times to events. For example, an airbag has to be inflated neither too earlier nor too late. Also, in the worst-case analysis of multi-hop networks, jitter plays a role, requiring a complementary best-case analysis. This chapter is based on Verduzco et al. (2017), a Work-in-Progress paper in which we showed how to perform a relative best-case response time analysis for Credit-Based Shaping. To the best of our knowledge, this work Verduzco et al. (2017) is the first to conduct research on the best-case analysis of Ethernet AVB and Ethernet TSN, in particular, using the relative response time analysis approach.

For the analysis, we introduce a notion of *a burst of frames* such that all frames in uninterfered schedules can be grouped into separate bursts and two neighboring bursts are separated by some time slacks. Two crucial observations regarding bursts are that, firstly, there is a straightforward way to describe the transmission of a burst in an uninterfered schedule and, secondly, adding interference to a burst of frames cannot lead to earlier transmission of any of the frames in the burst. Instead, adding interference prior to a burst of frames, i.e., including the time slack in between the bursts, may lead to a build-up of credit that is not present in an uninterfered schedule, and thus contributes to the earlier transmission instead of delaying

the start of frame transmissions. This finding is essential for our best-case response time analysis.

8.2 Burstiness

In this section, we take a closer look at burstiness before moving on to the best-case response analysis. We start with a formal definition of burstiness. Because the burst is also subject to the Credit-Based Shaping mechanism, we use the system model presented in Chapter 3 and the fundamental analysis presented in Chapter 4 to figure out the characteristics of burstiness.

To begin, a formal definition of *active interval* for Credit-Based Shaping is presented below in order to prepare for a formal definition of *a burst of frames*.

Definition 8.1 (Active Interval for CBS). *Given an uninterfered execution u_0 , a priority class X is active at a time t if it has either pending load or negative credit. An active interval of X is a maximal interval during which X is active.*

Typically, an *active interval* of a priority class X is defined as the time interval $[t_s, t_e)$ such that the following holds:

$$Pending_X(t) > 0 \vee CR_X(t) < 0 \text{ for all } t \in [t_s, t_e). \quad (8.1)$$

Definition 8.2 (A Burst of Frames). *Given an uninterfered execution u_0 , a burst of frames is defined as a sequence of frames of the same priority class that arrive in the same active interval.*

Next, we observe that for any later frame x_{i+1} in the burst, the arrival time has to be no later than the recovery time of its previous frame, i.e., $a^{u_0}(x_{i+1}) \leq s^{u_0}(x_i) + C(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+})$. Otherwise, there is no pending load and negative credit in the interval $[s^{u_0}(x_i) + C(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}), a^{u_0}(x_{i+1})]$, which contradicts the definition of a burst of frames. Applying this in Property 4.1 of Credit-Based Shaping (see page 29), we derive the following for bursts of frames.

Property 8.1. *Given a burst of frames in an uninterfered execution u_0 , and considering the frame arrivals as a sequence x_i , then the start times are determined by*

$$\begin{aligned} s^{u_0}(x_0) &= a^{u_0}(x_0), \\ s^{u_0}(x_{i+1}) &= s^{u_0}(x_i) + C^{u_0}(x_i) \cdot (1 + \frac{\alpha_X^-}{\alpha_X^+}). \end{aligned} \quad (8.2)$$

Based on this property, we can apply induction on the start times to find the following characterization of an uninterfered schedule for a burst.

Corollary 8.1. *Given a burst of frames in an uninterfered execution u_0 , and considering the arrivals of frames as a sequence x_i , we find for all $x_0, \dots, x_{i+k} \in X$:*

$$s^{u_0}(x_{i+k}) = s^{u_0}(x_i) + \sum_{j=i}^{i+k-1} C^{u_0}(x_j) \cdot \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right) \quad (8.3)$$

In the following section, we will look at how the scheduling of bursts may be accelerated compared to Corollary 8.1 when interference is introduced into the system.

8.3 Relative Best-Case Response Time

Before presenting a relative best-case response time analysis, we first introduce the definition of the best-case response time of a frame x given a set of possible executions \mathbb{U} :

$$BR^{\mathbb{U}}(x) = \inf_{u \in \mathbb{U}} f^u(x) - a^u(x). \quad (8.4)$$

Similar to what we did in the relative worst-case response time analysis, see Eq. (3.4), we can then derive a lower bound on the response time of a frame x in any execution u from that of u_0 by investigating the influence on start times as shown below:

$$\begin{aligned} & BR^{\mathbb{U}}(x) \\ &= \inf_{u \in \mathbb{U}} f^u(x) - a^u(x) \\ &= \inf_{u \in \mathbb{U}} s^u(x) + C^u(x) - a^u(x) \\ &= \inf_{u \in \mathbb{U}} s^u(x) + C^{u_0}(x) - a^{u_0}(x) \\ &= \inf_{u \in \mathbb{U}} s^{u_0}(x) + C^{u_0}(x) - a^{u_0}(x) + (s^u(x) - s^{u_0}(x)) \\ &\geq \left(\inf_{u \in \mathbb{U}} s^{u_0}(x) + C^{u_0}(x) - a^{u_0}(x) \right) + \left(\inf_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x)) \right) \\ &= BR^{U_0}(x) + \left(\inf_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x)) \right), \end{aligned} \quad (8.5)$$

where $BR^{U_0}(x)$ denotes the best-case response time of x in the set \mathbb{U}_0 of all uninterfered executions, and the term $\inf_{u \in \mathbb{U}} (s^u(x) - s^{u_0}(x))$ refers to the

relative best-case response time of a frame x . Note that we analyze the relative best-case response time without restricting a^u to any particular arrival pattern.

In the remainder of this section, we will seek a lower bound on this relative best-case response time for a stream of interest with priority class M . In particular, we restrict the stream M to be a series of frames that meet the criteria of *A Burst of Frames* (see Definition 8.2). Note that it is conceivable to generalize these results to avoid this restriction. However this involves further proof work that has not yet been realized in this dissertation.

The proof to obtain the lower bound of relative best-case reaction time for a burst of frames begins now.

Theorem 8.1. *Given a burst of n frames associated with priority class $M \in \mathbb{P}$, i.e., m_1, \dots, m_n in an uninterfered schedule u_0 . Consider the interference as the set $\mathbb{H} = \{X \in \mathbb{P} \mid X > M\}$ of all high-priority classes, and the set $\mathbb{L} = \{X \in \mathbb{P} \mid X < M\}$ of all low-priority classes. Assume that M and \mathbb{H} are under Credit-Based Shaping, and $\alpha_{\mathbb{H}}^+ + \alpha_M^+ \leq BW$. Then, for any execution u with associated uninterfered execution u_0 , any frame m_i in such a burst can at most be scheduled earlier by*

$$s^{u_0}(m_i) - s^u(m_i) \leq \min \left(\sum_{1 \leq j < i} C(m_j) \cdot \frac{\alpha_M^-}{\alpha_M^+}, \max \left(I_M^{\max} - C_M^{\min} \cdot \frac{\alpha_M^-}{\alpha_M^+}, 0 \right) \right). \quad (8.6)$$

Here, $C(m_j) = C^u(m_j) = C^{u_0}(m_j)$ denotes the transmission time of frame m_i , $I_M^{\max} = C_{\mathbb{L}}^{\max} \cdot (1 + \frac{\alpha_{\mathbb{H}}^+}{\alpha_{\mathbb{H}}^-}) + C_{\mathbb{H}}^{\max}$ denotes the maximum interference that a class M frame may experience.

Proof. Given a burst m_1, \dots, m_n , the start time of frame m_i in an uninterfered schedule is given by Corollary 8.1:

$$s^{u_0}(m_i) = s^{u_0}(m_1) + \sum_{1 \leq j < i} C(m_j) \cdot (1 + \frac{\alpha_M^-}{\alpha_M^+}). \quad (8.7)$$

The frame m_i only starts its transmission after each of the previous frames m_j with $1 \leq j < i$ finishes its transmissions and recover its credit.

For an interfered schedule, it is possible to build up credit, which can be then used to send frames continuously after one another (see Figure 3.2). We know that all the previous frames have to be transmitted prior to the start of frame m_i due to FIFO scheduling. Hence, a trivial lower bound for its start time is simply given by

$$s^u(m_i) \geq s^u(m_1) + \sum_{1 \leq j < i} C(m_j). \quad (8.8)$$

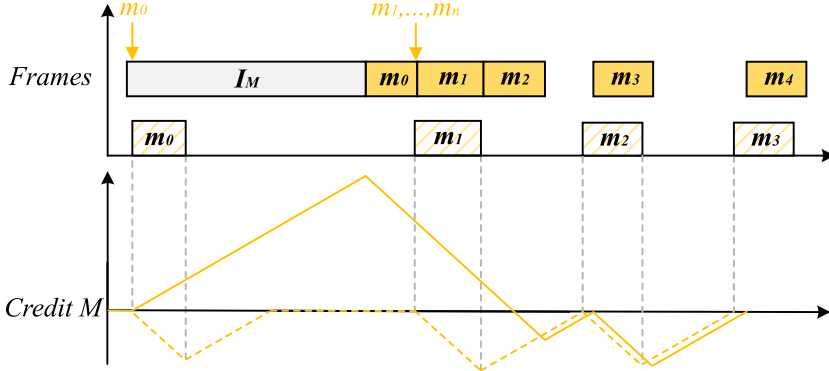


Figure 8.1: In this figure, we show that it is also possible that frames experience a negative delay with respect to the uninterfered case. Bounding this negative delay leads to a relative best-case response time. Note that frame m_0 before the burst m_1, \dots, m_n experiences some interference allowing to build up some credit before the burst starts.

Next, observe that a frame that experiences interference and is delayed by I_M compared to the uninterfered schedule, builds up a credit of $I_M \cdot \alpha_M^+$. If such a frame is part of a burst in the uninterfered schedule, then the credit that is built up may be used to send the next frames continuously till the credit is depleted. In particular, no recovery time will be needed in a time interval of length $I_M \cdot \frac{\alpha_M^+}{\alpha_M^-}$. The amount of recovery time that is present in the uninterfered schedule and is not considered in the interfered schedule is therefore $I_M \cdot \frac{\alpha_M^+}{\alpha_M^-} \cdot \frac{\alpha_M^-}{\alpha_M^+} = I_M$. The gained recovery time equals the suffered interference.

At this point, we conclude that any credit that leads to earlier transmission of frames must be built up before the start of the burst in the uninterfered schedule. Figure 8.1 shows an example. As can be seen, there is initially some positive credit $CR_M(s^u(m_1))$ in the interfered schedule when the burst m_1, \dots, m_n starts, allowing the first frames to be transmitted without recovery time. Frames are transmitted continuously without recovery time till the credit is depleted. Hence, the amount of recovery time that is removed in the interfered schedule is $CR_M(s^u(m_1)) \cdot \frac{1}{\alpha_M^+}$. The remaining frames after the credit is depleted will be scheduled with recovery time. Hence, a second lower bound for $s^u(m_i)$ is

$$s^u(m_i) \geq s^u(m_1) + \sum_{1 \leq j < i} C(m_j) \cdot \left(1 + \frac{\alpha_M^-}{\alpha_M^+}\right) - CR_M(s^u(m_1)) \cdot \frac{1}{\alpha_M^+}. \quad (8.9)$$

Based on the previous lower bounds, the start time of a frame m_i when

having credit $CR_M(s^u(m_1))$ is bounded by

$$s^u(m_i) \geq s^u(m_1) + \max\left(\sum_{1 \leq j < i} C(m_j), \sum_{1 \leq j < i} C(m_j) \cdot \left(1 + \frac{\alpha_M^-}{\alpha_M^+}\right) - CR_M(s^u(m_1)) \cdot \frac{1}{\alpha_M^+}\right). \quad (8.10)$$

Finally, observe that the first frame in an uninterfered burst can never be scheduled earlier in the interfered case, because credit needs pending load to build up. This gives us $s^{u_0}(m_1) \leq s^u(m_1)$. For any subsequent frame m_i , the amount of time that m_i is scheduled earlier is bounded as follows:

$$\begin{aligned} & s^{u_0}(m_i) - s^u(m_i) \\ & \leq s^{u_0}(m_1) + \sum_{1 \leq j < i} C(m_j) \cdot \left(1 + \frac{\alpha_M^-}{\alpha_M^+}\right) \\ & \quad - \left(s^u(m_1) + \max\left(\sum_{1 \leq j < i} C(m_j), \sum_{1 \leq j < i} C(m_j) \cdot \left(1 + \frac{\alpha_M^-}{\alpha_M^+}\right) - CR_M(s^u(m_1)) \cdot \frac{1}{\alpha_M^+}\right)\right) \\ & = s^{u_0}(m_1) - s^u(m_1) + \min\left(\sum_{1 \leq j < i} C(m_j) \cdot \frac{\alpha_M^-}{\alpha_M^+}, CR_M(s^u(m_1)) \cdot \frac{1}{\alpha_M^+}\right) \\ & \leq \min\left(\sum_{1 \leq j < i} C(m_j) \cdot \frac{\alpha_M^-}{\alpha_M^+}, CR_M(s^u(m_1)) \cdot \frac{1}{\alpha_M^+}\right). \end{aligned} \quad (8.11)$$

It only remains to bound $CR_M(s^u(m_1))$, i.e., to derive the maximum credit that can be built up just before a burst starts its transmission. To derive this, there should exist one frame m_0 , before the burst, and not a part of the burst, that allows to build up credit as much as possible (see Figure 8.1). As this frame has to be transmitted prior to the burst, it will deplete some of the credit and this depletion has to be as small as possible. Therefore, we assume that m_0 has a minimum frame size and the amount of credit at the start can be bounded as:

$$CR_M^{\max}(s^u(m_1)) = \max(CR_M(s^u(m_0)) - C_M^{\min} \cdot \alpha_M^-, 0). \quad (8.12)$$

In Theorem 5.4 it is shown that the maximum amount of credit that can be built up at any time for the priority class M , assuming $\alpha_M^+ + \alpha_H^+ \leq BW$, is given by $I_M^{\max} \cdot \alpha_M^+$, where $I_M^{\max} = C_L^{\max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{\max}$. Therefore, the maximum credit that can be built up before m_1 is transmitted is bounded by

$$CR_M^{\max}(s^u(m_1)) = \max(I_M^{\max} \cdot \alpha_M^+ - C_M^{\min} \cdot \alpha_M^-, 0). \quad (8.13)$$

The maximum amount of time that a frame m_i in a burst can be scheduled earlier in an interfered schedule compared to its uninterfered schedule is therefore

$$\begin{aligned} & \min \left(\sum_{1 \leq j < i} C(m_j) \cdot \frac{\alpha_M^-}{\alpha_M^+}, CR_M^{\max}(s^u(m_1)) \cdot \frac{1}{\alpha_M^+} \right) \\ &= \min \left(\sum_{1 \leq j < i} C(m_j) \cdot \frac{\alpha_M^-}{\alpha_M^+}, \max(I_M^{\max} - C_M^{\min} \cdot \frac{\alpha_M^-}{\alpha_M^+}, 0) \right), \end{aligned} \quad (8.14)$$

which concludes our proof. \square

From Theorem 8.1 and Eq.(8.5), it follows that, given the best-case response time $BR^{\mathbb{U}_0}(m_i)$ of a class M frame in a burst of frames in an uninterfered schedule, the best-case response time in an interfered schedule is the following:

$$\begin{aligned} BR^{\mathbb{U}}(m_i) &\geq BR^{\mathbb{U}_0}(m_i) \\ &\quad - \min \left(\sum_{1 \leq j < i} C(m_j) \cdot \frac{\alpha_M^-}{\alpha_M^+}, \max(I_M^{\max} - C_M^{\min} \cdot \frac{\alpha_M^-}{\alpha_M^+}, 0) \right). \end{aligned} \quad (8.15)$$

8.4 Conclusion

In this chapter, we presented and proved the relative best-case response time of a burst of frames under Credit-Based Shaping compared to its response time in a schedule without interference. In particular, we showed that frames in a burst in an uninterfered schedule can be scheduled earlier in an interfered schedule when a previous frame not belonging to the burst experiences interference and thus accumulates positive credit. Furthermore, this positive credit at the beginning of the burst allows the first few frames to be scheduled continuously without recovery time. As a result, rather than a delay, frames are transmitted relatively earlier.

Chapter 9

Reservation Strategy

9.1 Introduction

In this chapter, we represent the work of Cao et al. (2018a) to deal with the bandwidth allocation in time sensitive networks. In addition to the formerly introduced worst-case and best-case response time analysis, allocating network bandwidth for a distributed data-intensive embedded system is a crucial task. Over-reservation of resources generally leads to inefficiency in the system design, whereas under-reservation may result in a system failure. As a result, a systematic methodology to support allocating the bandwidth for applications is required.

In the Ethernet TSN standard, the bandwidth reservations for Stream Reservation traffic classes are provided through a Stream Reservation Protocol (SRP). This protocol allows a network designer to set bandwidth reservations for each individual priority class. It is suggested in the standard to reserve this bandwidth based on the expected utilization of that priority class (Clause 34.4 in IEEE (2014)). However, a bandwidth reservation analysis conducted in Ashjaei et al. (2017) shows that this suggestion may cause traffic to become unschedulable, while assigning a higher bandwidth may still help. The mechanism of Credit-Based Shaping is such that when multiple sources share the same priority level, a frame of one source can use the available credit and cause a frame of another source to be delayed until credit has recovered. Recovery of credit becomes faster when a higher bandwidth is allocated, so, while assigning a bandwidth at least equal to the utilization is necessary to avoid delays building up over time, the worst-case response time can often be decreased by choosing a higher bandwidth, allowing tighter deadline constraints to be met.

An algorithm is proposed in Ashjaei et al. (2017) to find a minimum bandwidth at each network link, at which the system is still schedulable. This work extends the schedulability analysis presented earlier in Bordoloi et al. (2014), which makes use of *Busy Period Analysis* to calculate the worst-case response times. As has been pointed out in Chapter 7, that analysis is often pessimistic due to the idling nature of the Credit-Based Shaping. To remedy this, the notion of *Eligible Interval* is introduced, resulting in a tight bound on the worst-case response times of Credit-Based Shaping, independent of the knowledge of the inter-priority traffic, except assumptions enforced by Ethernet TSN. Furthermore, the improvement in Bordoloi et al. (2014) to reduce pessimism is not considered in Ashjaei et al. (2017).

In this chapter, we use the relative WCRT analysis presented in Chapter 5 to determine a minimum bandwidth reservation for Credit-Based Shapers in a single switch, and achieve an improvement over the work presented in Ashjaei et al. (2017). Through rigorous proof, two constraints in determining a minimum bandwidth reservation are found, i.e., *deadline constraint* and *utilization constraint*, which clearly shows when the utilization bound suggested by the standard is sufficient for schedulability and when a higher allocation is necessary. In addition, we conduct a set of experiments and demonstrate an improvement in the bandwidth reservation efficiency, i.e., a decrease in the required bandwidth compared to Ashjaei et al. (2017) while maintaining the independence of the inter-priority interference. The derived formulas and the associated algorithms are of low complexity, and do not require an iterative calculation as in Ashjaei et al. (2017), of which the complexity depends on the system parameters.

The next section briefly describes earlier works related to bandwidth reservation in networks. Section 9.3 presents the main contribution of this chapter: the new bandwidth reservation analysis. The results of the experiments used to compare our analysis to the existing one are presented in Section 9.4. A discussion including the merits and demerits of the new analysis is conducted in Section 9.5 and finally Section 9.6 presents the conclusion and future work.

9.2 Related Works on Network Bandwidth Reservation

There are many techniques for realizing bandwidth reservation in networks, inspired mostly by resource reservation in processor domains. The main role of shapers is to regulate the traffic admitted to the network. Moreover, some real-time Ethernet protocols enforce a cyclic-based transmission in which

each cycle is divided into several transmission phases. This way, a transmission window is allocated to a set of messages, resembling server-based scheduling inherent in the processor domain. In addition, the hierarchical scheduling model has been used in the network domain, such as the hierarchical scheduling framework implemented by the FTT-SE protocol Iqbal et al. (2012) and the HaRTES architecture Santos et al. (2011). The main aim of the above techniques is to provide a policy to be able to reserve bandwidth for a certain message set.

In the scope of TSN, a Credit-Based Shaper and a time-aware shaper are defined that allow us to reserve bandwidth for each traffic class, which will be explained in the next section. However, not much attention has been paid to compute the minimum bandwidth to allocate for each traffic class in TSN prior to the work of Ashjaei et al. (2017); Cao et al. (2018a). The work in Ashjaei et al. (2017) proposes a technique to determine bandwidth for each traffic class according to the load. The work Cao et al. (2018a) presented in this chapter improves on that of Ashjaei et al. (2017) by using the tighter analysis of Cao et al. (2016a) (see Chapter 5).

9.3 Bandwidth Reservation Analysis

In this section, we address how to allocate the minimum network bandwidth for a given priority class under Credit-Based Shaping, such that all frames of that priority meet their deadlines. We restrict ourselves to three priorities, H (high), M (medium) and L (low), in order to compare our work with Ashjaei et al. (2017). We only study the bandwidth reservation for H and M , and assume that class L does not impose any timing constraints on its frame transmissions.

For a priority class or a stream X under Credit-Based Shaping, its network bandwidth is determined by the rising slope α_X^+ . Hence, to allocate the minimum network bandwidth means to find the minimum value of α_X^+ to fulfill the deadline requirement. Then, for a given priority X , we assume, just as we did in Chapter 7, that there are only a finite number (N_X) of periodic sources within priority X . A periodic *source* τ_i has its period T_i and a maximum transmission time C_i . It is required that every frame finishes its transmission across a single output port within an arbitrary deadline D_i . The total utilization $U_X = \sum_{\tau_i \in X} \frac{C_i}{T_i}$, is the sum of utilization of all sources in priority class X . In terms of other priority class Y , we only assume to know what the standard enforces, namely the maximum transmission time of that priority C_Y^{max} and the bandwidth reservation, i.e., the rising slope α_Y^+ .

Now we start to introduce a systematic analysis to allocate the bandwidth for streams H and M such that all periodic sources in streams H and M become schedulable, meaning that each source can fulfill its deadline in the worst-case scenario. We present the bandwidth reservation analysis for stream H before stream M , because the obtained reservation for stream H is required for the calculation of bandwidth for stream M .

9.3.1 Bandwidth Reservation for the High-Priority Stream

Before determining the minimum bandwidth reservation given the system model, we quickly go over the worst-case response analysis for a stream H . Referring to Corollary 5.2 (special case of Theorem 5.4), the worst-case response time of any source $\tau_i \in H$ is derived as follows:

$$WR^{\mathbb{U}}(\tau_i) = WR^{\mathbb{U}_0}(\tau_i) + C_{M,L}^{max}. \quad (9.1)$$

Note that $C_{M,L}^{max} = \max(C_M^{max}, C_L^{max})$.

We then apply Theorem 7.2 to calculate the worst-case response times in the uninterfered schedule. For a source $\tau_i \in H$, under the utilization bound $U_H \leq \frac{\alpha_H^+}{BW}$ and the reservation bound $\alpha_H^+ \leq BW$, the worst-case response time is calculated:

$$WR^{\mathbb{U}}(\tau_i) = \left(\sum_{\tau_j \in H, j \neq i} C_j \cdot \left(1 + \frac{\alpha_H^-}{\alpha_H^+}\right) \right) + C_i + C_{M,L}^{max}. \quad (9.2)$$

Based upon the worst-case response time analysis, the bandwidth reservation for stream H is further addressed in Theorem 9.1.

Theorem 9.1. *Given a stream H under Credit-Based Shaping, interfered by only low-priority streams M and L with maximum transmission times C_M^{max} and C_L^{max} . Assume that the stream H is only composed of periodic sources, and each source τ_i is characterized by its period T_i , maximum transmission time C_i , and deadline D_i . Then all sources in stream H are schedulable if for every source τ_i , we have $D_i > C_i + C_{M,L}^{max}$ and the bandwidth reservation of H satisfies ¹ :*

$$BW \geq \alpha_H^+ \geq \max \left(\frac{\sum_{\tau_j \in H, j \neq i} C_j}{\max_{\tau_i \in H} D_i - C_i - C_{M,L}^{max}}, U_H \right) \cdot BW. \quad (9.3)$$

¹If there exists only a single source τ_1 in the stream H , then this condition is $D_1 \geq C_1 + C_{M,L}^{max}$.

Proof. Firstly, in order to have finite worst-case response times, we need the necessary condition that the total utilization does not exceed the reservation, i.e., $U_H \leq \frac{\alpha_H^+}{BW}$. Hence, we have:

$$\alpha_H^+ \geq U_H \cdot BW . \quad (9.4)$$

Furthermore, we need to ensure the deadline requirement of every periodic source. If the deadline of each source can be met even in the worst-case scenario, i.e., $WR^U(\tau_i) \leq D_i$ for every $\tau_i \in H$, then it is guaranteed that all sources in stream H are schedulable. Based on the worst-case response time calculation in Eq. (9.2), and knowing $1 + \frac{\alpha_H^-}{\alpha_H^+} = \frac{BW}{\alpha_H^+}$, we obtain a lower bound of α_H^+ for a single source τ_i :

$$\alpha_H^+ \geq \frac{\sum_{\tau_j \in H, j \neq i} C_j \cdot BW}{D_i - C_i - C_{M,L}^{max}} . \quad (9.5)$$

Note that $D_i > C_i + C_{M,L}^{max}$ is also a necessary condition for schedulability. If it is not met, the presence of another source in H suffices to cause a deadline miss and τ_i is simply unschedulable.

Then, to satisfy all sources in stream H , we take the maximum over Eq. (9.5) for all $\tau_i \in H$ and combine with Eq. (9.4) to find the ultimate lower bound for α_H^+ as shown in the right-hand part of Eq. (9.3).

Lastly, we observe that α_H^+ should not exceed total network capacity, giving us the left-hand part of Eq. (9.3), and thus concluding the proof. \square

Algorithm 1 find α_H^+ for N_H sources in stream H

```

1: derive  $\alpha_H^+$  from Eq. (9.4)
2: for  $i = 1$  to  $N_H$  do
3:   derive newdc from Eq. (9.5)
4:    $\alpha_H^+ = \max(\alpha_H^+, \text{newdc})$ 
5: end for
6: if  $\alpha_H^+ > BW$  then
7:   return FALSE
8: else
9:   return  $\alpha_H^+$ 
10: end if
```

The essence of Theorem 9.1 is that the bandwidth reservation should satisfy two main constraints. One is the *utilization constraint*, stating that

the reservation should be at least the total utilization of all sources. The other is the *deadline constraint* to ensure that each source in the worst-case scenario meets its deadline. For a source τ_i , the numerator of the deadline constraint is the total time of all other sources in the same stream to transmit one frame and the denominator is the remaining time interval before the deadline when the transmission of a frame in τ_i and the maximum interfering time is taken out. The value of α_H^+ should always be larger than or equal to the maximum of these two constraints.

9.3.2 Bandwidth Reservation for the Medium-Priority Stream

For a stream M in the presence of mixed interference, we refer to Corollary 5.4 (one special case of Theorem 5.4) to calculate the worst-case response time of any source $\tau_i \in M$ as follows:

$$WR^{\mathbb{U}}(\tau_i) = WR^{\mathbb{U}_0}(\tau_i) + C_L^{max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{max} . \quad (9.6)$$

We then apply Theorem 7.2 to calculate the worst-case response times in the uninterfered schedule. For a source $\tau_i \in M$, under the utilization bound $U_M \leq \frac{\alpha_M^+}{BW}$ and the reservation bound $\alpha_H^+ + \alpha_M^+ \leq BW$, we have

$$\begin{aligned} WR^{\mathbb{U}}(\tau_i) = & \left(\sum_{\tau_j \in M, j \neq i} C_j \cdot \left(1 + \frac{\alpha_M^-}{\alpha_M^+}\right) \right) + C_i \\ & + C_L^{max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{max} . \end{aligned} \quad (9.7)$$

Theorem 9.2. *Given a stream M interfered by a high-priority stream H and a low-priority stream L , with M and H going through Credit-Based Shaping, and given respective maximum transmission times C_H^{max} , C_L^{max} for the interference, and a reservation α_H^+ for the high-priority stream. Assume the stream M is only composed of periodic sources, and each source τ_i is characterized by its period T_i , maximum transmission time C_i and deadline D_i . Then all sources in stream M are schedulable if for every source τ_i we have $D_i > C_i + C_L^{max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H^-}\right) + C_H^{max}$ and if the bandwidth reservation of*

M satisfies ²:

$$BW - \alpha_H^+ \geq \alpha_M^+ \geq \max \left(\frac{\sum_{\tau_j \in M, j \neq i} C_j}{\max_{\tau_i \in M} \frac{D_i - C_i - C_L^{max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H}\right) - C_H^{max}}, U_M}, U_M \right) \cdot BW . \quad (9.8)$$

The proof is analogous to that of Theorem 9.1. Once more, we find that the reservation should meet the utilization constraint:

$$\alpha_M^+ \geq U_M \cdot BW , \quad (9.9)$$

and the relative deadline constraint for each source $\tau_i \in M$:

$$\alpha_M^+ \geq \frac{\sum_{\tau_j \in M, j \neq i} C_j \cdot BW}{D_i - C_i - C_L^{max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H}\right) - C_H^{max}} . \quad (9.10)$$

Just note that this latter deadline constraint is a bit more complicated due to the influence of high-priority interference.

Algorithm 2 α_M^+ for N_M messages in stream M

```

1: derive  $\alpha_M^+$  from Eq. (9.9)
2: for  $i = 1$  to  $N_M$  do
3:   derive newdc from Eq. (9.10)
4:    $\alpha_M^+ = \max(\alpha_M^+, \text{newdc})$ 
5: end for
6: if  $\alpha_M^+ > BW - \alpha_H^+$  then
7:   return FALSE
8: else
9:   return  $\alpha_M^+$ 
10: end if
```

Note that to properly reserve bandwidth for stream M , one needs to know the reservation for stream H . As α_H^+ grows larger, more interference (worst-case relative delay) is introduced to stream M , see Theorem

²If there exists only a single source τ_1 in the stream M , then this condition is $D_1 \geq C_1 + C_L^{max} \cdot \left(1 + \frac{\alpha_H^+}{\alpha_H}\right) + C_H^{max}$.

5.4. Therefore, more reservation is needed for stream M to deal with the longer delay due to interference. Conversely, the minimum reservation of stream H consistently leads to the minimum reservation of stream M , refer to Eq. (9.10), which is in line with the objective to allocate the minimum bandwidth for streams H and M . Knowing the minimum value for α_H^+ is also crucial to the last step to determine the schedulability of stream M , since the calculated minimum value of α_M^+ can not exceed the remaining bandwidth excluding stream H .

The theorems and algorithms show that the new analysis inherits the merits of our WCRT analysis: the applicability to arbitrary deadlines, the low complexity as well as the independence. Next, we investigate how this new analysis for determining the minimum bandwidth reservations compares to the bandwidth reservation analysis outlined in Ashjaei et al. (2017).

9.4 Experiments

In this section, we perform a set of experiments to illustrate how to apply this new algorithm in determining bandwidth reservations for streams H and M . In all experiments, we compare the results of the new algorithm with those of the algorithm in Ashjaei et al. (2017). The results of the new algorithm is labeled with ‘EI-based Analysis’ in figures, while those of algorithm in Ashjaei et al. (2017) is labeled with ‘BP-based Analysis’.

We use identical sources for the first three experiments. Although the assumption of identical sources rarely occurs in practice, it provides valuable insights into how variables such as payloads influence the bandwidth reservation. In the first two experiments, we show the results of bandwidth reservation when the utilization varies, and we vary the utilization via gradually increasing the payload or the period of the sources. In the third experiment, as the new algorithm considers arbitrary deadlines, we show the influence of deadline upon the bandwidth reservation and compare with the algorithm Ashjaei et al. (2017) which considers constrained deadlines.

Subsequently, we use non-identical sources, which might be more practical compared to the previous settings, in the last experiment to investigate the schedulability ratios using the bandwidth reservation produced by the two algorithms.

Note that in all experiments we assume a total bandwidth of 100 Mbps and set the maximum payload for stream L according to the Ethernet standard, 1500 Bytes, and thus the maximum transmission C_L is 123.36 μ s. For stream M , we use the known maximum transmission time of the stream H

instead of the one from the standard. However, we still need less knowledge of the interference compared to Ashjaei et al. (2017) and remain independent of the detailed traffic model in stream H .

9.4.1 Identical Sources

In this subsection, we assume n identical periodic sources in stream H and n identical ones in stream M . Each source has a period T , a maximum transmission time C , and a deadline D .

We would like to show how bandwidth reservation needs to be adjusted when the utilization varies, and we vary the utilization via varying the payload or period in the first two experiments.

Experiment 1: Varying the Utilization via the Payload

In this experiment, the number of sources n is fixed to 4 for streams H and M , and each source has a fixed period and deadline $T = D = 1000$ μs . We vary the payload from 100 to 1500 Bytes with a step-size of 100 Bytes (the maximum transmission time C varies from 11.36 μs to 123.36 μs). The utilizations of streams H and M gradually increase to nearly 50%. The results of using the novel algorithm introduced in this dissertation and algorithm in Ashjaei et al. (2017) are represented in Figure 9.1.

The utilizations of streams H and stream M are the same, since the source sets are exactly the same. A linear increase of the utilization, represented by black dotted lines with crosses, is shown as the payload increases. The deadline constraint, see Eq. (9.5), is illustrated by blue dashed line with crosses. Our algorithm of bandwidth reservation takes the maximum of the utilization and deadline constraints, as shown in red circles. Moreover, we only show the results of bandwidth reservation if both stream M and stream L are schedulable.

For stream H , the deadline constraint gets closer to, but does not exceed, the utilization when the payload increases. For stream M , the deadline constraint is initially smaller than and surpasses the utilization when the payload continues to grow. Our algorithm alternates its choice of utilization constraint and deadline constraint. It is intuitive to see more reservation needed for stream M than stream L since stream M encounters more inter-priority interference.

The bandwidth reservation calculated by the algorithm in Ashjaei et al. (2017) is shown in cyan dash-dot line with triangles. For stream H , the reservation needed is slightly higher than the utilization, which does not deviate much from our algorithm. (The worst-case response time calculation

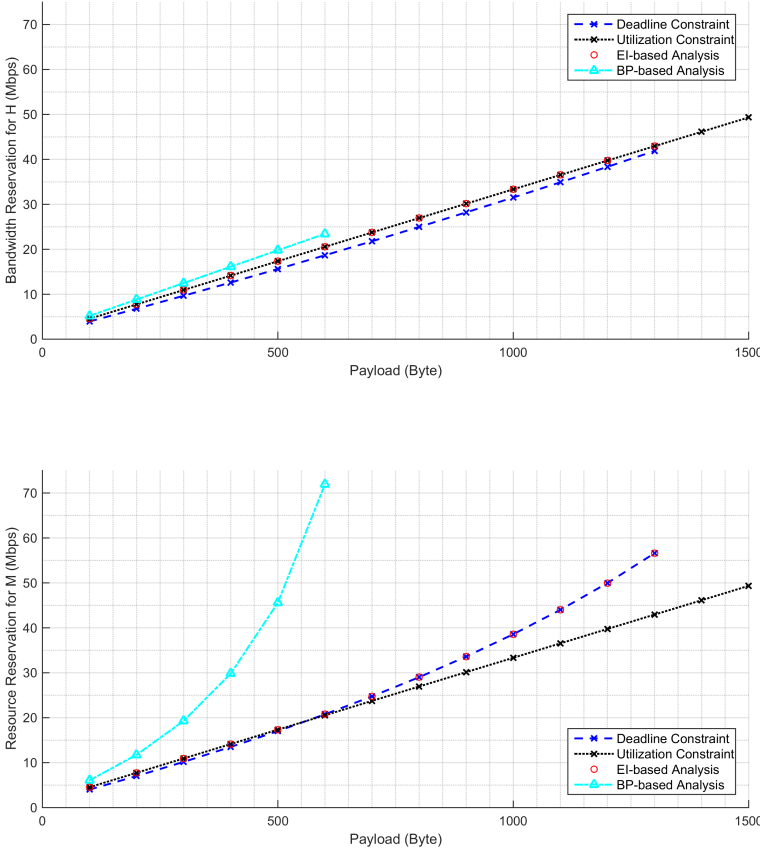


Figure 9.1: Bandwidth reservations, α_H^+ and α_M^+ , for varying payloads ($n = 4$ and $T = D = 1000 \mu s$).

in Ashjaei et al. (2017) needs to take the recovery time of the frame under analysis into account, refer to Eq. (14), while it is proven unnecessary in our independent analysis. This leads to a small deviation of these two approaches in the bandwidth reservation for stream H .)

For stream M , the reservation needed by Ashjaei et al. (2017) drastically increases when the payload increases. The last schedulable point given by Ashjaei et al. (2017) is when the payload is 600 Bytes and the utilization is 20.54%, while our algorithm allocates slightly higher than the utilization constraint 20.76 Mbps and Ashjaei et al. (2017) needs 71.95 Mbps. Meanwhile, the last schedulable point in our algorithm is when the payload is 1300 Bytes and the utilization reaches 42.94%, and it is still schedulable at a reservation of 56.60 Mbps. Note that if stream M is not schedulable, it is claimed that the source set of H and M is not schedulable.

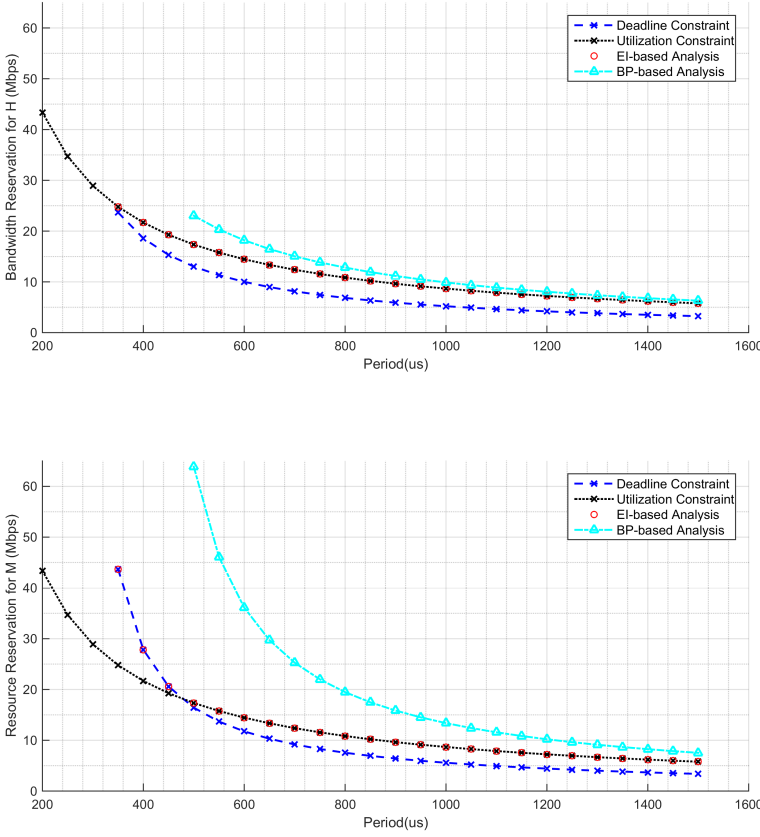


Figure 9.2: Bandwidth reservations, α_H^+ and α_M^+ , for varying periods ($n = 2$, $C = 43.36 \mu s$ and $D = T$).

The comparison above shows that our algorithm can schedule the source set at a relatively low reservation and improve the schedulability when the utilization is high. The relative inefficiency in bandwidth allocation for Ashjaei et al. (2017) can be explained by the fact that busy period analysis is pessimistic when applied to an idling server, such as a Credit-Based Shaper. This has already been intensively discussed in Chapter 6.

Experiment 2: Varying the Utilization via the Period

In this experiment, the number of sources n is fixed to 2 for streams H and M , and each source has a fixed payload of 500 Bytes, $C = 43.36 \mu s$, and the deadline equals to its period $D = T$. We vary the period from 200 to 1500 μs with a step-size of 100 μs . In Figure 9.2, the results of bandwidth reservation using two algorithms are shown.

The utilization decreases when the period increases. For stream H ,

the utilization constraint remains larger than the deadline constraint, and for stream M , the deadline constraint is at first larger than the utilization when the period is short and falls below the utilization when the period continues to grow. Our algorithm takes the maximum of the utilization and the deadline constraints.

Consistent with the previous experiment, the algorithm in Ashjaei et al. (2017) demands a higher reservation for both stream H and stream M than our algorithm. Especially for stream M , Ashjaei et al. (2017) demands much more bandwidth than ours when the utilization increases, which leads to unschedulability while our algorithm can still schedule the source set. Ashjaei et al. (2017) can schedule the source set when the period is 500 μs or higher, and our algorithm can schedule the source set when the period is 350 μs or higher.

In the two experiments above, the utilization is varied via either the payload or the period. In Figure 9.1 and 9.2, it firstly is shown how the deadline constraint and utilization constraint alternate the dominance in determining the bandwidth reservation in our algorithm. Although setting the bandwidth to the utilization constraint, as described in the IEEE 802.1Q standard, is sometimes sufficient, in other cases it is not, as indicated by the higher deadline constraint. Secondly, although the utilization constraints for streams H and M are the same, we observe that stream M requires a higher bandwidth reservation than stream H due to the presence of high-priority interference. Thirdly, our algorithm has been shown in these experiments to be more efficient in bandwidth reservation compared to the algorithm in Ashjaei et al. (2017), i.e., it requires less bandwidth reservation to schedule the source set. This efficiency improvement is more prominent in allocating bandwidth especially for stream M . Hence, some source sets that are not schedulable according to Ashjaei et al. (2017) become schedulable in our algorithm.

As explained in the system model, our analysis allows arbitrary deadlines. However, the solution proposed in Ashjaei et al. (2017) is based on the work in Bordoloi et al. (2014), which assumes constrained deadlines for their worst-case response time analysis. Our analysis has its advantage over Ashjaei et al. (2017) in its applicability when the deadline is larger than its period. We set up the experiment to see the deadline's influence upon bandwidth reservation.

Experiment 3: Arbitrary Deadline vs. Constrained Deadline

The number of sources n is fixed to 4, and each source has a fixed period $T = 1500 \mu\text{s}$ and a fixed payload of 300 Bytes, $C = 27.36 \mu\text{s}$. The utilization

is fixed and intentionally set to a low value to investigate the bandwidth reservation when the deadline is short. We vary the deadline from 150 to 2700 μs with a step-size of 150 μs .

For comparison in this experiment, we keep the deadline equal to its period when deadline is larger than its period for Ashjaei et al. (2017), which is a safe but pessimistic assumption. The results of bandwidth reservation using the two algorithms are presented in Figure 9.3.

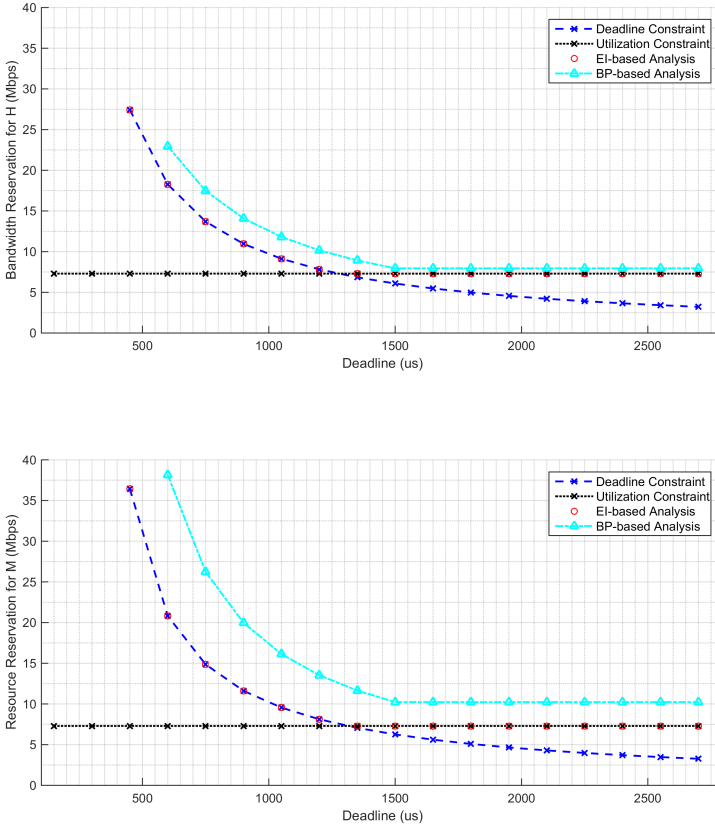


Figure 9.3: Bandwidth reservations, α_H^+ and α_M^+ , of varying deadlines ($n = 4$, $T = 1500 \mu\text{s}$ and $C = 27.36 \mu\text{s}$).

The utilization constraint remains at a value of 7.30 Mbps. Our algorithm alternates its choice of utilization and deadline constraints for streams H and M . When the deadline increases, the utilization becomes more dominant in determining the bandwidth reservation. In particular, when the deadline is larger than the period, our algorithm takes the utilization constraint.

As shown in Figure 9.3, the algorithm in Ashjaei et al. (2017) needs a higher reservation than our algorithm when the deadline is smaller than or equal to its period. Note that the deviation between the two algorithms is not significantly large as the utilization is intentionally kept low to allow for a wider range of deadline variation. When the deadline is larger than the period, the bandwidth reservation of Ashjaei et al. (2017) remains constant. It is over-estimated due to the stricter assumption to keep the deadline equal to its period. However, the bandwidth estimate will not be improved much in this experiment set-up since it also needs to satisfy the utilization constraint.

9.4.2 Non-identical Sources

In this subsection, we investigate the schedulability of non-identical sources using the bandwidth reservation produced by the two algorithms when the utilization gradually increases. We assume non-identical sources in this experiment to challenge the schedulability of the set. However, we do not fully randomize the source sets because FIFO and non-preemptive transmission results in low schedulability in general. If we fully randomize the process to generate the source sets, the majority are not schedulable, making it meaningless to compare the two algorithms. Therefore, we only randomize the payload and keep the utilization evenly-distributed among different sources of streams H and M .

Experiment 4: Schedulability Test of Non-identical Sources

We assume n non-identical periodic sources in stream H and n non-identical ones in stream M . The total utilization is evenly distributed among $2 \cdot n$ sources, i.e., if the total utilization is U , then each source has a utilization of $\frac{U}{2 \cdot n}$. In this experiment, n is fixed to 5, and the total utilization increases from 10% to 90%. For each source, the payload is randomly chosen from 200 to 600 Bytes, and the period changes accordingly to keep the utilization the same. For each utilization, we generate the source set and run the experiment for 10000 times. The schedulability ratio, i.e., the fraction of the source sets that are schedulable, is recorded and shown in Figure 9.4.

The schedulability ratio of the new algorithm is presented by the red dashed line with circles, and the one in Ashjaei et al. (2017) is illustrated by the cyan dash-dot line with triangles. Both ratios decrease when the utilization increases. Our algorithm consistently outperforms the algorithm in Ashjaei et al. (2017), i.e., it has a higher or equal to schedulability ratio than Ashjaei et al. (2017) for the same utilization. Moreover, we observed that the source sets, which are schedulable by Ashjaei et al. (2017), are always schedulable by the new algorithm in this experiment.

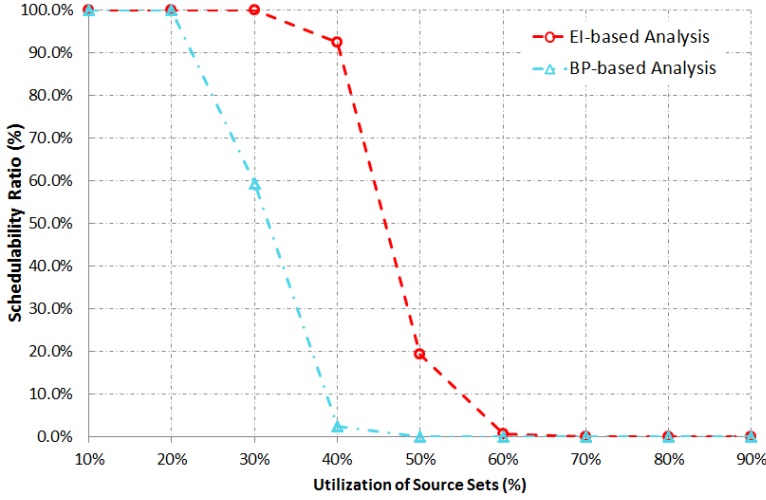


Figure 9.4: Schedulability test of non-identical source sets.

This result is consistent with the previous three experiments, the new algorithm improves the schedulability compared to Ashjaei et al. (2017) due to its efficiency in the bandwidth reservation.

9.5 Discussion

In this section, we will firstly elaborate more on the essence of the new analysis, and then discuss the merits and the demerits compared to busy-period based analysis in Ashjaei et al. (2017).

9.5.1 Utilization Constraint vs. Deadline Constraint

Our new analysis mainly imposes two constraints on the minimum bandwidth reservation: the *utilization constraint* and the *deadline constraint*, refer to Eq. (9.3) and Eq. (9.8). These two constraints can alternate dominance as shown in the experiments. Hence, setting the bandwidth to the utilization constraint, as described in the IEEE 802.1Q standard, is not always sufficient. This is consistent with the findings of Ashjaei et al. (2017). However, as the efficiency improves, some source sets previously claimed unschedulable by Ashjaei et al. (2017) using the utilization constraint become schedulable in our new analysis.

9.5.2 Merits

This bandwidth reservation analysis is derived from our relative WCRT analysis (see Chapter 5) and thus inherits its merits. It is applicable to arbitrary deadlines, whereas its counterpart in Ashjaei et al. (2017) is only applicable to constrained deadlines. The analysis has a linear complexity, which is considerably lower than the pseudo-polynomial complexity of the counterpart. It is independent of the detailed traffic model of the interference while the counterpart is. It is worth noting that if we do not know the inter-priority traffic except assumptions enforced by the standard, this new analysis provides a tight bound on the bandwidth reservation, which is robust to changes in traffic flow in the interference. When the detailed traffic model of the inter-priority interference is provided, pessimism arises in our analysis. In this case, independence is gained at the expense of efficiency; however, in the experiments we performed, the new analysis still gives better results than the existing analysis, but this may not always be the case, depending on whether the new analysis method can exploit detailed knowledge of the interference.

9.5.3 Demerits

Pessimism arises in the new analysis when we assume the independence of the inter-priority traffic while a detailed traffic model is known. There are two sources of pessimism.

The first source of pessimism lies in estimating the intermediate burst. According to Corollary 5.4, the worst-case relative delay is $C_L^{max} \cdot (1 + \frac{\alpha_H^+}{\alpha_H}) + C_H^{max}$. When there at least exists one single source in stream H , we know the worst-case relative delay is at least $C_L^{max} + C_H^{max}$. The intermediate high-priority burst given by $C_L^{max} \cdot \frac{\alpha_H^+}{\alpha_H}$ may not always be feasible, which gives rise to the pessimism in the analysis. Adding the detailed traffic model of interference may improve the efficiency in the bandwidth reservation. However, as we explained in Chapter 6, it is not preferable to lose the simplicity and independence of the analysis for small improvements.

The second source of pessimism is due to using the value from the standard for the maximum transmission time C_X^{max} . It may be reduced to improve the efficiency by using the known maximum transmission time of all sources to be scheduled in a stream for C_X^{max} . Despite of the reduced independence, the independence of the detailed traffic model is still kept. We leave it to the network designers to balance the choices between the efficiency and the independence.

In none of the experiments we performed, the analysis in Ashjaei et al.

(2017) outperforms the new analysis. As the experiments rule out the second source of pessimism, we now discuss the potential causes why the first source of pessimism does not lead to inferiority. The origin of the first source of pessimism, i.e., the intermediate high-priority burst, depends on two parameters, the bandwidth reservation for stream H , i.e., α_H^+ , and the maximum transmission time of stream L , i.e., C_L^{max} . In this work, we aim at finding the minimum α_H^+ given a set of H sources. Moreover, the maximum transmission times C_L^{max} is restricted to the standard, further limit the pessimism in the new analysis. Finally, the counterpart based on busy period analysis has been pointed out in Chapter 7 to be more pessimistic in most cases. Given the two sources of pessimism in the new analysis, we do not yet know whether the new analysis always outweighs the existing one. It is worth noting that, despite the pessimism, the independence of our analysis is still highly desirable in order to make the WCRT estimate robust against changes in the system design and robust in the event that the system fails to perform according to the specified traffic model.

9.6 Conclusion and Future Work

In this chapter, we proposed a new bandwidth reservation strategy for Ethernet TSN for Credit-Based Shaping without considering scheduled traffic. The analysis is based on our relative WCRT analysis that uses *Eligible Intervals* (see Chapter 5). Given the model of an output port of a single switch, i.e., streams H and M undergo Credit-Based Shaping while stream L does not undergo shaping, the bandwidth reservation is determined in two steps. The first step is to find the minimum bandwidth reservation for stream H composed of periodic sources (according to Theorem 9.1 and Algorithm 1) and then the minimum bandwidth reservation for stream M (according to Theorem 9.2 and Algorithm 2). Note that we assume the periodic traffic model only for the stream under analysis and we do not have any assumption of the traffic model for inter-priority interference.

In brief, the analysis imposes two constraints on the bandwidth reservation: the *utilization constraint* and the *deadline constraint*. It is mandatory to set the bandwidth reservation at least to the utilization constraint. However, setting the bandwidth to the utilization constraint, as described in the IEEE 802.1Q standard, is not always sufficient. More bandwidth is required if the deadline constraint is dominant.

As this analysis is based on our relative WCRT analysis using *Eligible Intervals*, it inherits the merits: the linear complexity, the applicability to arbitrary deadlines, the independence of the inter-priority interference except assumptions enforced by the standard and the efficiency in band-

width reservation resulting from the tightness. Meanwhile, it also inherits the demerits: the pessimism due to lack of the detailed traffic model of the interference and the pessimism using the value from the standard instead of the known maximum of the stream for the maximum transmission time C_X^{max} .

This new analysis improves the existing one in Ashjaei et al. (2017) based on *Busy Period Analysis* in terms of complexity, applicability and independence. In this work, we observed that the new analysis always outperforms Ashjaei et al. (2017) regarding the efficiency based on the experiment setups in which the second source of pessimism is excluded. The question whether the new analysis always outperforms Ashjaei et al. (2017) in terms of efficiency remains unanswered and we leave it for future research.

A follow-up work could be to extend the current bandwidth reservation analysis from single-hop to multi-hop. In terms of inter-priority interference, our relative WCRT analysis can readily scale to multi-hop. The challenge lies in the WCRT analysis of Credit-Based Shaping for a single priority in a multi-hop network. Also, in Ethernet TSN other shapers are considered in combination with Credit-Based Shaping to guarantee low latency for control traffic, such as time-aware shaping standardized in IEEE 802.1Qbv to support scheduled traffic. We would like to deal with this heterogeneity into the scheduling and extend the current analysis to support the scheduled traffic.

Chapter 10

Conclusions and Future Work

10.1 Discussion of Main Results

In this dissertation, we investigated the communication challenge of sophisticated cyber-physical systems that typically use heterogeneous traffic over networks to connect their various components. These different types of traffic impose different timing requirements on the network architecture of the system. Because current real-time network standards typically do not provide sufficient bandwidth to keep up with the growing needs in the cyber-physical systems domain, the Ethernet TSN working group has been established. This working group intends to develop standards that provide real-time guarantees on top of the well-known and widely available high-bandwidth Ethernet standard. In Ethernet TSN, every traffic stream is assigned a priority and given a traffic-shaping strategy based on its real-time delivery requirements. The shaping strategies make a first step towards achieving real-time guarantees, but how to quantitatively estimate the time bounds, particularly worst-case as well as best-case bounds, has been a challenge for engineers and scientists.

The research focus of this dissertation has been to provide a new methodology to deal with timing analyses in the context of Ethernet TSN. In particular, our research efforts emphasize the tightness and independence of the analysis of so-called Credit-Based Shapers. As far as we are aware, independence - i.e., the robustness of the analysis against changes in the environment - has not received much attention. Most previous research works in the field assume a rather strict periodic or sporadic traffic model, also and in particular for the interfering traffic, see e.g. Bordoloi et al. (2014);

Axer et al. (2014). Consequently, minor environmental changes that lead to minor changes in the interfering traffic can easily render an analysis invalid. On the one hand, having more information about interfering traffic in our traffic model may seem advantageous since this enables us to find better time bounds in theory. On the other hand, depending on this information makes the analysis less robust against changes in a systems design and requires a revisit of the analysis every time a change is made. This is in particular cumbersome when a large system is composed of components supplied by third parties.

We devised a new method for handling heterogeneous traffic under Credit-Based Shaping. By dividing the traffic into two categories - those with the same priority as the traffic stream of interest and those with other priorities - the timing analysis for Credit-Based Shaped traffic can be broken down into a basic timing analysis of a single priority plus a relative timing analysis of the change in timing caused by all other priorities. We show that the basic timing analysis becomes a reasonably straightforward analysis that can be performed using standard techniques from real-time systems. More importantly, the subsequent relative timing analysis can be carried out using only information regarding the parameters of the Credit-Based Shapers and without information regarding the precise traffic that flows through these shapers. As a result, a network designer just has to know the detailed traffic of the priority of interest and does not need to know more than the shaper parameters of other priorities. This leads to the preliminary isolation of design concerns.

In order to achieve an analysis that is not only independent but also as tight as possible, we introduce the notion of Eligible Interval in this dissertation, tailored to Credit-Based Shaping. In a Credit-Based Shaper, the traffic transmission depends on a level of credit determined by shaping rules explained in the Ethernet TSN standard. As a consequence, the behavior of the Credit-Based Shaper is that of an idling server and is, therefore, less suitable for the usual busy period analysis. An Eligible Interval of a priority class is a maximal interval during which that priority is eligible for transmission; in other words, not only is there pending traffic, but the shaping rules also allow for that traffic to be transmitted. With the support of Eligible Intervals, we can establish a connection between the schedules with interference and the corresponding uninterfered schedule.

The main theorems of this dissertation show how the credit level in a Credit-Based Shaper is proportional to the relative delay experienced by a traffic stream during an Eligible Interval. This allows us to perform the split, as mentioned above, between a basic timing analysis and a relative timing analysis. The worst-case analysis subsequently comes down to analyzing the

maximum amount of credit that can build up during an Eligible Interval. This worst-case analysis is tight in the presence of a single high-priority interference, as we show how to construct a traffic instance that indeed builds up the computed amount of credit. Furthermore, the analysis turns out to be considerably more straightforward to execute than the earlier analyses presented in Bordoloi et al. (2014); Axer et al. (2014). Even more surprisingly, in actual comparisons, our estimated results outperform those of the other two existing works in most cases, even though the other works use more information on the precise interference.

A subsequent mathematical analysis of the best-case response time revealed a surprising finding, as presented in Chapter 8. When a frame prior to the burst, but not part of it, experiences the maximum interference and thus accumulates positive credit, this positive credit enables the first few frames of the burst to be scheduled continuously without recovery time. As a result, the presence of interference may cause the first few frames of a burst to be transmitted *earlier* than in an uninterfered scenario, resulting in a negative relative best-case response time.

The last contribution of this dissertation is a new systematic bandwidth reservation strategy for Ethernet TSN for Credit-Based Shaping. It is based on our independent relative worst-case response time analysis method with no assumption on the traffic model for inter-priority interference. The analysis reveals two constraints on bandwidth reservation: the utilization constraint and the deadline constraint. Our findings in Chapter 9 show when setting the bandwidth reservation to the utilization according to the IEEE 802.1Q standard is sufficient and when the deadline constraint is more dominant.

10.2 Limitations and Future Work

As this is the first work that employs Eligible Interval Analysis for a relative, rather than direct, response time analysis, the method has certain limitations that may be lifted by future research. One limitation is that the developed independent timing analysis works only for Credit-Based Shaping. With the knowledge of the shaping properties of hybrid interference, we can examine how the interference affects the basic interference-free schedules in a relative manner. Several works have already taken advantage of our relative worst-case response times to deal with the timing analysis with Time Aware Shaper on top of Credit-Based Shaper Hassani et al. (2020); Maxim and Song (2017). We furthermore expect that the technique of performing a relative timing analysis using tailored Eligible Intervals can be extended to other shapers and perhaps to other types of idling services in

general, but further investigations are needed to solidify those claims.

A second limitation of our current work is that the analysis only considers the shaping of traffic in a single hop in a network route. Calculating the relative worst-case response times over multiple hops is rather straightforward, which requires simply the summation of relative worst-case response times at each hop. The challenges stem from the basic analysis of a single priority over multiple hops. Credit-Based Shaping disperses traffic to achieve an average bandwidth over an extended period. Without interference, a stream undergoing Credit-Based Shaping at one hop, probably does not need to be shaped and further delayed at a second hop. However, cumulative credit caused by inter-priority interference can transform scattered traffic into a burst. How to adequately describe the output of one hop, which is crucial as the input of the next hop, is a challenge that must be overcome for basic timing analysis.

A third possible improvement is that we currently use a periodic traffic model in the basic analysis to determine the bounds of multiple streams that all have the same priority to compare our results with those of other works. A more realistic assumption following usage in practice might improve the applicability of the overall timing analyses. For instance, video traffic undergoing Credit-Based Shaping is commonly characterized as bursty rather than periodic traffic. Periodically generated images form rather large payloads, which are split over a number of Ethernet frames and transmitted as a burst of frames arriving shortly after one another. One of the goals of Credit-Based Shaping is to reduce burstiness in the network, which means that it is especially interesting to study how an arriving burst is deformed by it. A continuation of our work within the European Penta HiPER project <https://penta-eureka.eu/project-overview/penta-call-3/hiper/> indicates that, with our independent analysis approach, we can easily update the basic analysis with bursty traffic to gain insight into response times at the level of images rather than only at the level of individual frames.

In summary, shaping in Ethernet TSN aims to manage network traffic in a way that provides predictable timing bounds on its stream while not exceeding its permitted bandwidth to ensure the timing guarantees for the low priorities. Our quantitative research reveals the effect of regulating traffic with Credit-Based Shaping. Through our independent timing analysis and bandwidth reservation strategy, we can quantitatively determine how much the Credit-Based Shaping of one priority contributes as interference to other priorities with only traffic information on the shaper level. In this sense, Credit-Based Shaping serves its purpose of regulating traffic. For future research involving more shaping strategies, an important criterion for success will be whether it is possible to find again an independent timing

analysis and independent bandwidth reservation for the additional shapers. We expect that such an evaluation will be possible and will enable engineers to determine the best shaping method and parameters for a system configuration and provide general feedback on shaping performance to the standardization committees.

Bibliography

- Ashjaei M, Patti G, Behnam M, Nolte T, Alderisi G, Bello LL (2017) Schedulability Analysis of Ethernet Audio Video Bridging Networks with Scheduled Traffic Support. *Real-Time Systems* 53(4):526–577, DOI 10.1007/s11241-017-9268-5
- Axer P, Thiele D, Ernst R, Diemer J (2014) Exploiting Shaper Context to Improve Performance Bounds of Ethernet AVB Networks. In: *Proc. 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp 1–6, DOI 10.1145/2593069.2593136
- Bordoloi UD, Aminifar A, Eles P, Peng Z (2014) Schedulability Analysis of Ethernet AVB Switches. In: *Proc. 20th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp 1–10, DOI 10.1109/RTCSA.2014.6910530
- Bril R, Lukkien J, Verhaegh W (2009) Worst-Case Response Time Analysis of Real-Time Tasks Under Fixed-Priority Scheduling with Deferred Preemption. *Real-Time Systems* 42(1):63–119, DOI 10.1007/s11241-009-9071-z
- Cao J (2016) The Matlab Codes that Generate Experiment Results for Comparison. URL <https://tue.data.surfsara.nl/index.php/s/cCZYHw6Abr5DAWC>
- Cao J, Cuijpers PJJ, Bril RJ, Lukkien JJ (2016a) Independent yet Tight WCRT Analysis for Individual Priority Classes in Ethernet AVB. In: *Proc. 24th International Conference on Real-Time Networks and Systems (code)*, pp 55–64, DOI 10.1145/2997465.2997493
- Cao J, Cuijpers PJJ, Bril RJ, Lukkien JJ (2016b) Tight Worst-Case Response-Time Analysis for Ethernet AVB Using Eligible Intervals. In: *Proc. 12th IEEE World Conference on Factory Communication Systems (WFCS)*, pp 1–8, DOI 10.1109/WFCS.2016.7496507

- Cao J, Ashjaei M, Cuijpers PJJ, Bril RJ, Lukkien JJ (2018a) An Independent yet Efficient Analysis of Bandwidth Reservation for Credit-Based Shaping. In: Proc. 14th IEEE International Workshop on Factory Communication Systems (WFCS), pp 1–10, DOI 10.1109/WFCS.2018.8402345
- Cao J, Cuijpers PJJ, Bril RJ, Lukkien JJ (2018b) Independent WCRT analysis for individual priority classes in Ethernet AVB. *Real-Time Systems* 54(4):861–911, DOI 10.1007/s11241-018-9321-z
- Cuijpers P, Bril R (2006) Towards Periodic Budgeting in Real-Time Calculus. Tech. rep., Technische Universiteit Eindhoven
- Davis R, Burns A, Bril R, Lukkien J (2007) Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised. *Real-Time Systems* 35(3):239–272, DOI 10.1007/s11241-007-9012-7
- De Azua JAR, Boyer M (2014) Complete Modelling of AVB in Network Calculus Framework. In: Proc. 22nd International Conference on Real-Time Networks and Systems (RTNS), DOI 10.1145/2659787.2659810
- Diemer J, Rox J, Ernst R (2012a) Modeling of Ethernet AVB Networks for Worst-Case Timing Analysis. In: Proc. 7th Vienna International Conference on Mathematical Modelling (MATHMOD), pp 848–853, DOI 10.3182/20120215-3-AT-3016.00150
- Diemer J, Thiele D, Ernst R (2012b) Formal Worst-Case Timing Analysis of Ethernet Topologies with Strict-Priority and AVB Switching. In: Proc. 7th IEEE International Symposium on Industrial Embedded Systems (SIES), pp 1–10, DOI 10.1109/SIES.2012.6356564
- Hassani H, Cuijpers PJJ, Bril RJ (2020) Work-in-Progress: Layering Concerns for the Analysis of Credit-Based Shaping in IEEE 802.1 TSN. In: Proc. 16th IEEE International Conference on Factory Communication Systems (WFCS), pp 1–4, DOI 10.1109/WFCS47810.2020.9114408
- Henia R, Hamann A, Jersak M, Racu R, Richter K, Ernst R (2005) System Level Performance Analysis - the SymTA/S Approach. *Computers and Digital Techniques* 152(2):148–166, DOI 10.1049/ip-cdt:20045088
- IEEE (2005) IEEE 802.1 Audio/Video Bridging Task Group. URL <http://www.ieee802.org/1/pages/avbridges.html>
- IEEE (2012) IEEE 802.1 Time-Sensitive Networking Task Group. URL <http://www.ieee802.org/1/pages/tsn.html>

- IEEE (2014) IEEE Std. 802.1Q-2014, Standard for Local and Metropolitan Area Networks, Bridges and Bridged Networks
- IEEE (2016) IEEE Standard 802.1Qbv, Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks Amendment 25: Enhancements for Scheduled Traffic
- Imtiaz J, Jasperneite J, Han L (2009) A Performance Study of Ethernet Audio Video Bridging (AVB) for Industrial Real-Time Communication. In: Proc. 14th IEEE Conference on Emerging Technologies Factory Automation (ETFA), pp 1–8, DOI 10.1109/ETFA.2009.5347126
- Iqbal Z, Almeida L, Marau R, Behnam M, Nolte T (2012) Implementing Hierarchical Scheduling on COTS Ethernet Switches Using a Master/Slave Approach. In: Proc. 7th IEEE International Symposium on Industrial Embedded Systems (SIES), DOI 10.1109/SIES.2012.6356572
- Leboudec J, Thiran P (2001) Network Calculus. Springer Berlin, Heidelberg, DOI 10.1007/3-540-45318-0
- Lee KC, Lee S, Lee MH (2006) Worst Case Communication Delay of Real-Time Industrial Switched Ethernet with Multiple Levels. IEEE Transactions on Industrial Electronics 53(5):1669–1676, DOI 10.1109/TIE.2006.881986
- Lehoczky J (1990) Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. In: Proc. 11th Real-Time Systems Symposium (RTSS), pp 201–209, DOI 10.1109/REAL.1990.128748
- Li X, George L (2017) Deterministic Delay Analysis of AVB Switched Ethernet Networks Using an Extended Trajectory Approach. Real-Time Systems 53(1):121–186, DOI 10.1007/s11241-016-9260-5
- Martin S, Minet P (2006) Schedulability Analysis of Flows Scheduled with FIFO: Application to the Expedited Forwarding Class. In: Proc. 20th IEEE International Parallel Distributed Processing Symposium (IPDPS), DOI 10.1109/IPDPS.2006.1639424
- Maxim D, Song YQ (2017) Delay Analysis of AVB Traffic in Time-Sensitive Networks (TSN). In: Proc. 25th International Conference on Real-Time Networks and Systems (RTNS), pp 18–27, DOI 10.1145/3139258.3139283
- Reimann F, Graf S, Streit F, Glas M, Teich J (2013) Timing Analysis of Ethernet AVB-based Automotive E/E Architectures. In: Proc. 18th IEEE Conference on Emerging Technologies Factory Automation (ETFA), pp 1–8, DOI 10.1109/ETFA.2013.6648024

- Santos R, Behnam M, Nolte T, Pedreiras P, Almeida L (2011) Multi-level Hierarchical Scheduling in Ethernet Switches. In: Proc. 11th International Conference on Embedded Software (EMSOFT), DOI 10.1145/2038642.2038671
- Thangamuthu S, Concer N, Cuijpers P, Lukkien J (2015) Analysis of Ethernet-Switch Traffic Shapers for In-Vehicle Networking Applications. In: Proc. Design, Automation and Test in Europe Conference (DATE), pp 55–60, DOI 10.7873/DATE.2015.0045
- Thiele D, Ernst R (2016) Formal Worst-Case Timing Analysis of Ethernet TSN's Burst-Limiting Shaper. In: Proc. Design, Automation and Test in Europe Conference (DATE), pp 187–192, DOI 10.3850/9783981537079_0276
- Thiele D, Ernst R, Diemer J (2015) Formal Worst-Case Timing Analysis of Ethernet TSN's Time-Aware and Peristaltic Shapers. In: Proc. IEEE Vehicular Networking Conference (VNC), pp 251–258, DOI 10.1109/VNC.2015.7385584
- Verduzco HR, Cuijpers PL, Cao J (2017) Work-in-Progress: Best-Case Response Time Analysis for Ethernet AVB. In: Proc. 38th Real-Time Systems Symposium (RTSS), pp 378–380, DOI 10.1109/RTSS.2017.00043
- Wandeler E, Thiele L, Verhoef M, Lieverse P (2006) System architecture evaluation using modular performance analysis: A case study. International Journal on Software Tools for Technology Transfer 8(6):649–667, DOI 10.1007/s10009-006-0019-5

Summary

There is an increasing demand in the automotive industry, high-tech systems, and industrial automation, to accommodate heterogeneous data exchange with high bandwidth communication systems capable of providing real-time guarantees for time-sensitive traffic. The development on top of Ethernet standards enables prioritized data transmission and relies on traffic shaping techniques to allow for different priorities to meet their respective bandwidth and latency needs. We focus on one standardized Credit-Based Shaper that aims to provide the required latency and throughput for audio/video traffic.

Most timing analyses with respect to traffic shaping require a detailed traffic model at each priority level. This dependence on traffic models complicates the analysis, and developers confront the problem of an undesired coupling between different applications, often built by different development teams. Moreover, these analyses are not robust against changes in the system design and are not robust in the event that the system fails to perform according to the specified traffic model.

In this work, we invent an independent timing analysis that considers all traffic from other priorities as inter-priority interference and merely requires shaper-level knowledge rather than a detailed traffic model. Unlike conventional methods, which calculate the worst-case response time of the traffic of interest in a single attempt, we take a relative approach to achieve the same goal, by first computing the worst-case response time in an un-interfered execution and then adding on the relative worst-case response time. This independent analysis not only leads to a preliminary simplification of design concerns, but the resulting estimate can also be proven tight under certain conditions, in contrast to the conventional busy period analyses. We attribute the tightness of our analysis to our introduction of the notion of Eligible Intervals. Unlike busy period analyses, Eligible Intervals are tailored to take the idling nature of Credit-Based Shaping into account.

This dissertation consists of three main parts. Firstly, we present a rela-

tive worst-case response time analysis for a Credit-Based Shaper of interest, given the interference consisting of multiple high priorities under Credit-Based Shaping and multiple low priorities. We provide rigorous proof to determine an upper bound where only shaper-level knowledge is relevant, followed by a further investigation of the conditions under which this independent analysis is tight. We also compare our relative worst-case time analysis to the conventional busy period analyses. It turns out that certain sources of pessimism are present in the conventional approaches, which can be remedied by the use of our relative worst-case response time analysis.

Secondly, we use this independent timing analysis to estimate the relative best-case response time. In particular, we show that adding interference prior to a burst of frames in an uninterfered execution contributes to an earlier transmission instead of a delay in the start of frame transmissions.

Thirdly, we use this independent timing analysis as a foundation to determine a minimum bandwidth reservation for Credit-Based Shapers, outperforming previous work based on busy period analyses. Two constraints in determining a minimum bandwidth reservation are derived, namely a deadline constraint and a utilization constraint. In addition, we conduct a set of comparative experiments and demonstrate an improvement in the bandwidth reservation efficiency, i.e., a decrease in the required bandwidth while retaining the independence of the inter-priority interference.

Curriculum Vitae

Jingyue Cao was born in Ningbo, China, on May 1, 1984. She earned her Bachelor's degree in Biomedical Engineering from Zhejiang University in 2008. Then she moved to the Netherlands to pursue a Master's degree in Electrical Engineering at Eindhoven University of Technology. Her research experience began with her first internship at NXP Semiconductors, where she worked on the instantaneous localization and classification of subtitles in TV applications. Afterwards, her research interests evolved towards biosignal processing, and she conducted research on estimating core body temperature using multivariate sensor data during her Master's project at Philips Research in 2010.

After completing her master's degree, Jingyue continued her work at Philips Research as a signal processing designer. She conducted research on diverse projects, such as developing algorithms for predicting baby awakening and crying using unobtrusive sensor data or estimating respiration rate from photoplethysmography signals collected from wrist-worn devices.

In 2014, she joined the System Architecture and Networking group (later transformed into the Interconnected Resource-aware Intelligent Systems group) in the Mathematics and Computer Science Department at Eindhoven University of Technology as a Ph.D. candidate. Her research focused on addressing communication challenges in the Control based on Data-Intensive Sensing project under the STW's Robust Design of Cyber-Physical Systems program. This collaborative effort involved two other Ph.D. candidates who tackled control and communication challenges, respectively, as well as industrial partners FEI (later acquired by Thermo Fisher Scientific) and Technolution. Jingyue's work resulted in the publication of three conference papers and one Work-in-Progress paper. One of her papers at RTNS 2016 received an outstanding paper award, and an extended version was invited for publication in the Real-Time Systems journal. This dissertation presents the results from the publications mentioned above.

Titles in the IPA Dissertation Series since 2020

M.A. Cano Grijalba. *Session-Based Concurrency: Between Operational and Declarative Views.* Faculty of Science and Engineering, RUG. 2020-01

T.C. Nägele. *CoHLA: Rapid Co-simulation Construction.* Faculty of Science, Mathematics and Computer Science, RU. 2020-02

R.A. van Rozen. *Languages of Games and Play: Automating Game Design & Enabling Live Programming.* Faculty of Science, UvA. 2020-03

B. Changizi. *Constraint-Based Analysis of Business Process Models.* Faculty of Mathematics and Natural Sciences, UL. 2020-04

N. Naus. *Assisting End Users in Workflow Systems.* Faculty of Science, UU. 2020-05

J.J.H.M. Wulms. *Stability of Geometric Algorithms.* Faculty of Mathematics and Computer Science, TU/e. 2020-06

T.S. Neele. *Reductions for Parity Games and Model Checking.* Faculty of Mathematics and Computer Science, TU/e. 2020-07

P. van den Bos. *Coverage and Games in Model-Based Testing.* Faculty of Science, RU. 2020-08

M.F.M. Sondag. *Algorithms for Coherent Rectangular Visualiza-*

tions. Faculty of Mathematics and Computer Science, TU/e. 2020-09

D. Frumin. *Concurrent Separation Logics for Safety, Refinement, and Security.* Faculty of Science, Mathematics and Computer Science, RU. 2021-01

A. Bentkamp. *Superposition for Higher-Order Logic.* Faculty of Sciences, Department of Computer Science, VU. 2021-02

P. Derakhshanfar. *Carving Information Sources to Drive Search-based Crash Reproduction and Test Case Generation.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2021-03

K. Aslam. *Deriving Behavioral Specifications of Industrial Software Components.* Faculty of Mathematics and Computer Science, TU/e. 2021-04

W. Silva Torres. *Supporting Multi-Domain Model Management.* Faculty of Mathematics and Computer Science, TU/e. 2021-05

A. Fedotov. *Verification Techniques for xMAS.* Faculty of Mathematics and Computer Science, TU/e. 2022-01

M.O. Mahmoud. *GPU Enabled Automated Reasoning.* Faculty of Mathematics and Computer Science, TU/e. 2022-02

M. Safari. *Correct Optimized GPU Programs.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2022-03

M. Verano Merino. *Engineering Language-Parametric End-User Programming Environments for DSLs.* Faculty of Mathematics and Computer Science, TU/e. 2022-04

G.F.C. Dupont. *Network Security Monitoring in Environments where Digital and Physical Safety are Critical.* Faculty of Mathematics and Computer Science, TU/e. 2022-05

T.M. Soethout. *Banking on Domain Knowledge for Faster Transactions.* Faculty of Mathematics and Computer Science, TU/e. 2022-06

P. Vukmirović. *Implementation of Higher-Order Superposition.* Faculty of Sciences, Department of Computer Science, VU. 2022-07

J. Wagemaker. *Concurrent Separation Logics for Safety, Refinement, and Security.* Faculty of Science, Mathematics and Computer Science, RU. 2022-08

R. Janssen. *Refinement and Partiality for Model-Based Testing.* Faculty of Science, Mathematics and Computer Science, RU. 2022-09

M. Laveaux. *Accelerated Verification of Concurrent Systems.* Faculty of Mathematics and Computer Science, TU/e. 2022-10

S. Kochanthara. *A Changing Landscape: On Safety & Open Source in Automated and Connected Driving.* Faculty of Mathematics and Computer Science, TU/e. 2023-01

L.M. Ochoa Venegas. *Break the Code? Breaking Changes and Their Impact on Software Evolution.* Faculty of Mathematics and Computer Science, TU/e. 2023-02

N. Yang. *Logs and models in engineering complex embedded production software systems.* Faculty of Mathematics and Computer Science, TU/e. 2023-03

J. Cao. *An Independent Timing Analysis for Credit-Based Shaping in Ethernet TSN.* Faculty of Mathematics and Computer Science, TU/e. 2023-04

