

# Network Calculus

Steffen Bondorf<sup>\*1</sup>, Anne Bouillard<sup>\*2</sup>, Markus Fidler<sup>\*3</sup>,  
Jörg Liebeherr<sup>\*4</sup>, and Lisa Maile<sup>†5</sup>

1 Ruhr-Universität Bochum, DE. [steffen.bondorf@rub.de](mailto:steffen.bondorf@rub.de)

2 Huawei Technologies – Boulogne-Billancourt, FR. [anne.bouillard@huawei.com](mailto:anne.bouillard@huawei.com)

3 Leibniz Universität Hannover, DE. [markus.fidler@ikt.uni-hannover.de](mailto:markus.fidler@ikt.uni-hannover.de)

4 University of Toronto, CA. [jorg@ece.utoronto.ca](mailto:jorg@ece.utoronto.ca)

5 Universität Erlangen-Nürnberg, DE. [maile@ida.ing.tu-bs.de](mailto:maile@ida.ing.tu-bs.de)

---

## Abstract

Network calculus is a versatile method for analysing queueing systems with applications in Internet Quality of Service (QoS), wireless networks, Ethernet with delay guarantees, real-time systems, and feedback control. Using min-plus or max-plus algebra and deterministic or stochastic bounds, this Dagstuhl Seminar aims to bring together the deterministic and stochastic network calculus community to discuss recent research, future directions, and collaboration.

The modelling power of network calculus allows it to represent different systems, making it applicable to a wide variety of queueing problems. Thus, it has been proposed in various contexts for new emerging technologies such as IEEE Time Sensitive Networking (TSN), IETF Deterministic Networking (DetNet), and 5G Ultra-Reliable Low Latency Communications (URLLC), with important applications in factory automation, aerospace onboard, and automotive in-vehicle networks.

The two communities of deterministic and stochastic network calculus have grown closer in recent years, for example, as deterministic network calculus results have been incorporated into stochastic network calculus, demonstrating the need for and value of strong collaboration between the communities.

Recent developments in network calculus algorithms include modular and optimisation approaches, parallelizable methods that improve performance bounds, machine learning techniques, but also the adaptation of network calculus for design automation and system configuration.

This report documents the programme, the new contributions, and the results of Dagstuhl Seminar 24141 “Network Calculus”.

**Seminar** April 1–4, 2024 – <https://www.dagstuhl.de/24141>

**2012 ACM Subject Classification** Computer systems organization → Real-time systems; Networks → Network performance analysis; Networks → Network performance modeling

**Keywords and phrases** age of information, effective bandwidths, network calculus, performance evaluation, queueing network

**Digital Object Identifier** 10.4230/DagRep.14.4.1

---

\* Editor / Organizer

† Editorial Assistant / Collector



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Network Calculus, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 1–22

Editors: Steffen Bondorf, Anne Bouillard, Markus Fidler, and Jörg Liebeherr



DAGSTUHL  
REPORTS Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Executive Summary

*Lisa Maile (Universität Erlangen-Nürnberg, DE)*

License  Creative Commons BY 4.0 International license  
© Lisa Maile

Our Dagstuhl Seminar brought together leading experts in the field of network calculus to discuss recent research activities, identify future directions, and establish or strengthen cooperation. The seminar fostered collaborations and new ideas by focusing on diversity, incorporating perspectives from research and industry, varying levels and areas of expertise, and participants of different positions and ages. Attendees came from various scientific disciplines, including deterministic and stochastic network calculus, Age-of-Information, industrial communication, and wireless technologies. All combined under the need for Quality of Service (QoS) requirements. Participants contributed scientific presentations and workshops on their current and future research.

Network calculus is a theoretical framework for analyzing networks, mainly, but not limited to communication networks, but applications are also possible in the field of energy systems or even emergency call centers. By modeling data flows and systems as mathematical functions, network calculus allows the calculation of guaranteed (deterministic/stochastic) upper bounds. This analysis can be applied to various elements, including individual hops or entire systems. Additionally, network calculus has been used for certification purposes in avionic networks, such as the Airbus A380 [1], demonstrating its importance and applicability in designing safety-critical systems that require stringent QoS performance guarantees.

The seminar focused on several main areas: integrating algebras and performance metrics, exploring network topology and parallel systems, applying network calculus to emerging technologies, and developing algorithms and tools. New insights in these topics were presented, followed by group work on individual problems. These group sessions explored algorithmic challenges, mathematical and computational complications, and practical applications of network calculus.

In the context of algebras and performance metrics, discussions centered on representing network calculus as a systems theory under min-plus algebra, as well as integrating min-plus and max-plus algebras. This dual approach could provide a more comprehensive framework for analyzing complex network behaviors. A significant challenge addressed was the issue of rare perturbations, where deterministic network calculus currently lacks the ability to quantify the frequency or rarity of worst-case scenarios. Participants discussed the need for more nuanced metrics to distinguish between frequent low delays and rare high delays.

The seminar also highlighted the powerful modeling capabilities of network calculus in representing various systems, such as links, traffic shapers, and scheduling policies, which can be composed into arbitrary topologies. Advanced results from deterministic network calculus, like pay bursts and multiplexing only once phenomena, have recently been incorporated into stochastic network calculus, necessitating strong cooperation between the deterministic and stochastic communities. Parallel systems, particularly in the context of multi-path transport and fork-join models, were another central point, with discussions addressing the complexities and opportunities inherent in these configurations.

Emerging technologies such as Time-Sensitive Networking (TSN), Deterministic Networking (DetNet), and Ultra-Reliable Low Latency Communications (URLLC) were also a significant focus of the seminar. Network calculus currently receives a lot of research attention due to these emerging topics, which attract high interest from both industry and academia. These technologies are crucial for applications in factory automation, aerospace

onboard systems, and automotive in-vehicle networks, where performance guarantees are critical. The seminar emphasized the integration of network calculus with these emerging technologies to ensure robust performance analysis and guarantees, supporting applications that meet stringent reliability and latency requirements.

Recent advancements in network calculus algorithms were another key theme of the seminar. Participants discussed combining modular and optimization approaches and developing highly parallelizable methods that iteratively improve performance bounds. The use of machine learning for appropriate decomposition in modular approaches was highlighted, allowing for more efficient and scalable solutions to complex network performance challenges. Additionally, the development of tools to rapidly disseminate novel results and facilitate extensive community-based research artifact evaluation and reproducibility verification was emphasized.

Discussions following each presentation helped identify open problems and challenges to be addressed by the community. Participants highlighted the importance of collaboration between the different network calculus communities and experts, the topics which are still open and where our community could focus in the future, and potentials to make network calculus more accessible to the public.

The seminar also included connecting personal and scientific discussions during a slightly rainy but enjoyable hike, long extended dinners, and quizzes to engage in team building. These social events provided additional opportunities for participants to engage and exchange ideas.

In summary, the Dagstuhl Seminar on network calculus facilitated significant discussions on advancing the field through interdisciplinary collaboration, integrating new technologies, and leveraging new methodologies for optimization. The insights and outcomes from the seminar pave the way for future research and development, ensuring network calculus receives recognition and remains a robust and versatile tool for network analysis and performance guarantees.

## References

- 1 F. Francés, C. Fraboul, and J. Gried, *Using Network Calculus to optimize the AFDX network*, ERTS 2006: 3rd European Congress ERTS Embedded real-time software, 2006.

## 2 Table of Contents

### Executive Summary

<i>Lisa Maile</i> . . . . .	2
-----------------------------	---

### Overview of Talks

Quasi-Deterministic Burstiness Bound for Aggregate of Independent, Periodic Flows <i>Anne Bouillard</i> . . . . .	6
Dynamics of energy storage systems with self-discharge <i>Almut Burchard</i> . . . . .	6
Automata-Theoretic Characterizations of Real-Time Calculus <i>Samarjit Chakraborty</i> . . . . .	7
Impact of AS6802 Synchronization Protocol on Time- Triggered and Rate-Constrained traffic <i>Anaïs Finzi</i> . . . . .	9
Exploiting Minimal Arrival Curves To Deal With Negative Service Curves <i>Anja Hamscher and Vlad-Cristian Constantin</i> . . . . .	9
Towards a Calculus for Wireless Networks: Recent Advances and What's Next <i>Yuming Jiang</i> . . . . .	10
Time Sensitive Networks and Network Calculus <i>Jean-Yves Le Boudec</i> . . . . .	10
Network Calculus Characterization of Congestion Control <i>Harvinder Lehal</i> . . . . .	11
Decentralized Reservation Protocols in Time-Sensitive Networking: Applying Network Calculus Without Central Network Overview <i>Lisa Maile</i> . . . . .	11
Statistical Age-of-Information Bounds for Parallel Systems <i>Mahsa Noroozi</i> . . . . .	12
Equivalent versions of Total Flow Analysis and SAIHU tool <i>Stéphan Plassart</i> . . . . .	12
Time-stationary and Event- based Age-of-Information: A Palm calculus bridge <i>Amr Rizk</i> . . . . .	13
Logic-Based Formal Analysis of Network Performance <i>Mina Tahmasbi Arashloo</i> . . . . .	13
Factors Limiting the Modularity of xTFA <i>Ludovic Thomas</i> . . . . .	14
Performance and Scaling of Parallel Systems with Blocking Start and/or Departure Barriers <i>Brenton Walker</i> . . . . .	14
Improving algorithms and software for DNC <i>Raffaele Zippo</i> . . . . .	15

**Working groups**

Network Calculus and Machine Learning/Artificial Intelligence <i>Michael Beck</i> . . . . .	15
Network Calculus versus other types of real-time network analysis <i>Anne Bouillard and Jörg Liebeherr</i> . . . . .	18
Core of the Network Calculus Theory <i>Marc Boyer</i> . . . . .	19
Network Calculus Tools <i>Lisa Maile and Anja Hamscher</i> . . . . .	19
<b>Participants</b> . . . . .	22

### 3 Overview of Talks

#### 3.1 Quasi-Deterministic Burstiness Bound for Aggregate of Independent, Periodic Flows

Anne Bouillard (*Huawei Technologies – Boulogne-Billancourt, FR*)

**License** © Creative Commons BY 4.0 International license  
© Anne Bouillard

**Joint work of** Hossein Tabatabaee, Anne Bouillard, Jean-Yves Le Boudec  
**Main reference** Seyed Mohammadhossein Tabatabaee, Anne Bouillard, Jean-Yves Le Boudec: “Quasi-Deterministic Burstiness Bound for Aggregate of Independent, Periodic Flows”, in Proc. of the Quantitative Evaluation of Systems – 20th International Conference, QEST 2023, Antwerp, Belgium, September 20-22, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14287, pp. 205–220, Springer, 2023.

**URL** [https://doi.org/10.1007/978-3-031-43835-6\\_15](https://doi.org/10.1007/978-3-031-43835-6_15)

Time-sensitive networks require timely and accurate monitoring of the status of the network. To achieve this, many devices send packets periodically, which are then aggregated and forwarded to the controller. Bounding the aggregate burstiness of the traffic is then crucial for effective resource management. In this paper, we are interested in bounding this aggregate burstiness for independent and periodic flows. A deterministic bound is tight only when flows are perfectly synchronized, which is highly unlikely in practice and would be overly pessimistic.

We compute the probability that the aggregate burstiness exceeds some value. When all flows have the same period and packet size, we obtain a closed-form bound using the Dvoretzky–Kiefer–Wolfowitz inequality. In the heterogeneous case, we group flows and combine the bounds obtained for each group using the convolution bound. Our bounds are numerically close to simulations and thus fairly tight. The resulting aggregate burstiness estimated for a non-zero violation probability is considerably smaller than the deterministic one: it grows in square root of  $n \log n$ , instead of  $n$ , where  $n$  is the number of flows.

#### 3.2 Dynamics of energy storage systems with self-discharge

Almut Burchard (*University of Toronto, CA*)

**License** © Creative Commons BY 4.0 International license  
© Almut Burchard

**Joint work of** Almut Burchard, Majid Raeis, Jörg Liebeherr

Energy storage is a crucial component of the smart grid, since it provides the ability to buffer transient fluctuations of the energy supply from renewable sources. Even without a load, energy storage systems experience a reduction of the stored energy through self-discharge. This talk presents analysis of the self-discharge phenomenon using a queueing system model, which we refer to as leakage queue. When the average net charge is positive, we discover that the leakage queue operates in one of two regimes: a leakage-dominated regime and a capacity-dominated regime. We find that in the leakage-dominated regime, the stored energy stabilizes at a point that lies well below the storage capacity. The predictions are validated in a numerical example where the energy supply resembles a wind energy source.

### 3.3 Automata-Theoretic Characterizations of Real-Time Calculus

Samarjit Chakraborty (University of North Carolina at Chapel Hill, US)

License © Creative Commons BY 4.0 International license  
© Samarjit Chakraborty

Real-Time Calculus (RTC) [2, 7] has proven to be a general framework for analyzing a variety of distributed and heterogeneous real-time systems that employ different scheduling policies. By drawing principles from the theory of Network Calculus, RTC uses *arrival curves* to model how computation and communication tasks in a real-time system are triggered, or in other words, the workload in a system. Similarly, *service curves* are used to model how these tasks are served, or the computation and communication bandwidth available to a task. With the output of a computation or communication task *triggering* a subsequent task on the same or a different resource, by using the theory of max plus and min plus algebra, the framework helps in analyzing how workload flows through a network of computation and communication resources and derives the timing properties of the tasks mapped onto these resources. With this analysis being compositional, RTC has been found to be effective in analyzing the timing properties of large networked embedded systems in scalable fashion.

The analysis in RTC uses algebraic techniques and is “functional” in nature, and does not allow the modeling of “state” in a straightforward manner. This talk discussed how timing properties of a cyber-physical system (CPS) can be modeled using principles of RTC, but using a corresponding automata-theoretic model. Representing an *arrival curve* as a finite automaton allows access to a rich set of automata-theoretic modeling and verification tools, including model checking. Currently, the control strategy in a CPS is determined independent of the characteristics of the platform running the software-implementation of the controller. The control strategy typically includes a sampling period and a required sensor-to-actuator delay that the engineer implementing the controller has to guarantee using suitable scheduling techniques. While such a design flow follows the principle of “separation of concerns” and distinguishes between a model and its implementation, it is increasingly turning out to be overly conservative. This is because the strict separation between design and implementation carries no information on how much deviation in the timing behavior determined at the controller design stage is safe. As a result, the notion of safety has been synonymous with “meeting all deadlines.” This is a difficult goal to meet, given the complexities of modern implementation platforms and the increasing volume of software code that is implemented on them in various domains such as automotive or robotics.

To address this, the talk outlined a notion of *safety*, where any trajectory of the closed-loop system (*i.e.*, plant + controller) in its state space is considered to be safe as long as it is contained within a predefined safety pipe around its nominal trajectory. Such a nominal trajectory could be the system trajectory when the controller is subjected to an ideal timing behavior (*i.e.*, it experiences no deadline misses). The question is to then determine which timing behaviors result in safe trajectories? For this, deadline hit/miss patterns are represented as RTC arrival curve-like *weakly-hard* constraints. Whether a chosen weakly-hard constraint is safe or not is determined through a safe but approximate reachability analysis of the closed-loop system [5, 10]. It is then noted that any weakly-hard constraint – that represents a set of binary strings capturing patterns of deadline hits or misses experienced by a control task – can be represented by an equivalent finite automaton. Since the union of regular languages is also regular, a collection of finite automata representing safe timing behaviors of the closed-loop system can be represented by a single finite automaton. Now, consider a set of controllers that need to be implemented on the same shared platform,

with their utilization being greater than 100% if *all* of their deadlines are to be met. Is it possible to schedule them such that each controller misses some deadlines, but are nevertheless *safe*, following the notion of safety outlined above?

The talk outlined a method to address this question. It showed that the product of the automaton corresponding to the safe timing behaviors of the individual closed-loop systems contains such schedules. More details on this may be found in [8, 9]. Checking whether a given weakly-hard constraint is safe, involves solving a reachability analysis problem, which is computationally very expensive. This was addressed by solving a safe but an *approximate* reachability analysis problem, which can be pessimistic, *i.e.*, return “unsafe” for weakly-hard constraints that are actually safe. This was addressed by using stochastic hypothesis testing based statistical verification techniques [4]. These are computationally less expensive and are less pessimistic, but provide parametrizable probabilistic guarantees.

A natural question that arises in the context of using automata-theoretic representations of RTC is: *why not use formalisms like timed automata?* This is because they have also proven to be very effective for the modeling and verification of real-time systems and are amenable to model checking. The talk concluded by discussing that RTC provides a “count-based abstraction” that is useful and sufficient in many setups. Here, it is sufficient to record *how many deadlines were missed?* Recording the precise times at which the individual deadlines were missed or when tasks were triggered – as would be done by formalisms like timed automata – would not be necessary. The schedule synthesis techniques outlined in [8, 9] from the automata representations of RTC-based weakly-hard constraints are also simpler than those that would be necessary if formalisms like timed automata would be used to represent the safe timing behaviors of the different controllers sharing computation and communication resources. Other automata-theoretic representations of RTC may be found in [1, 3, 6].

## References

- 1 A. Bouillard, L. T. X. Phan, and S. Chakraborty. Lightweight modeling of complex state dependencies in stream processing systems. In *15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009.
- 2 S. Chakraborty, S. Künzli, and L. Thiele. A general framework for analysing system properties in platform-based embedded system designs. In *Design, Automation and Test in Europe Conference and Exposition (DATE)*, 2003.
- 3 S. Chakraborty, L. T. X. Phan, and P. S. Thiagarajan. Event count automata: A state-based model for stream processing systems. In *26th IEEE Real-Time Systems Symposium (RTSS)*, 2005.
- 4 B. Ghosh, C. Hobbs, S. Xu, F. D. Smith, J. H. Anderson, P. S. Thiagarajan, B. Berg, P. S. Duggirala, and S. Chakraborty. Statistical verification of autonomous system controllers under timing uncertainties. *Real Time Syst.*, 60(1):108–149, 2024.
- 5 C. Hobbs, B. Ghosh, S. Xu, P. S. Duggirala, and S. Chakraborty. Safety analysis of embedded controllers under implementation platform timing uncertainties. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(11):4016–4027, 2022.
- 6 L. T. X. Phan, S. Chakraborty, and P. S. Thiagarajan. A multi-mode real-time calculus. In *29th IEEE Real-Time Systems Symposium (RTSS)*, 2008.
- 7 L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2000.
- 8 S. Xu, B. Ghosh, C. Hobbs, P. S. Thiagarajan, and S. Chakraborty. Safety-aware flexible schedule synthesis for cyber-physical systems using weakly-hard constraints. In A. Takahashi, editor, *28th Asia and South Pacific Design Automation Conference (ASPDAC)*, 2023.



- 9 S. Xu, B. Ghosh, C. Hobbs, P. S. Thiagarajan, P. Joshi, and S. Chakraborty. Safety-aware implementation of control tasks via scheduling with period boosting and compressing. In *29th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2023.
- 10 A. Yeolekar, R. Metta, C. Hobbs, and S. Chakraborty. Checking scheduling-induced violations of control safety properties. In *20th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 13505 of *Lecture Notes in Computer Science*. Springer, 2022.

### 3.4 Impact of AS6802 Synchronization Protocol on Time- Triggered and Rate-Constrained traffic

Anaïs Finzi (*TTTech Computertechnik – Wien, AT*)


License  Creative Commons BY 4.0 International license  
© Anaïs Finzi

TTEthernet is an Ethernet-based synchronized network technology compliant with the AFDX standard. It supports safety-critical applications by defining different traffic classes: Time-Triggered (TT), Rate-Constrained (RC), and Best-Effort traffic. The synchronization is managed through the AS6802 protocol, which defines so-called Protocol Control Frames (PCFs) to synchronize the local clock of each device. In this presentation, we analyze the synchronization protocol to assess the impact of the PCFs on TT and RC traffic.

We propose a method to decrease the impact of PCFs on TT and a new Network Calculus model to compute RC delay bounds with the influence of both PCF and TT traffic. We finish with a performance evaluation to i) assess the impact of PCFs, ii) show the benefits of our method in terms of reducing the impact of PCFs on TT traffic and iii) prove the necessity of taking the PCF traffic into account to compute correct RC worst-case delays and provide a safe system.

### 3.5 Exploiting Minimal Arrival Curves To Deal With Negative Service Curves

Anja Hamscher (*RPTU – Kaiserslautern, DE*) and Vlad-Cristian Constantin (*RPTU – Kaiserslautern, DE*)

License  Creative Commons BY 4.0 International license  
© Anja Hamscher and Vlad-Cristian Constantin

Joint work of Anja Hamscher, Vlad-Cristian Constantin, Jens B. Schmitt

Main reference Anja Hamscher, Vlad-Cristian Constantin, Jens B. Schmitt: “Extending Network Calculus To Deal With Partially Negative And Decreasing Service Curves”, *CoRR*, Vol. abs/2403.18042, 2024.


URL <https://doi.org/10.48550/ARXIV.2403.18042>

When considering multiple flow scenarios, strictness of service curves is required to obtain a residual service curve for a single flow. Without strictness, the residual service curve exhibits an interval over which it is negative and decreasing, i.e., it is not in  $\mathcal{F}_0^\uparrow$  anymore. However, this assumption is needed to calculate performance bounds. Introducing a lower bound on the arrivals to a system, the inherent issue can be avoided.

In our talk, we discuss the issue at hand and show how the existing performance bounds can be adjusted to work with service curves that are not in  $\mathcal{F}_0^\uparrow$ . We also discuss potential applications.

### 3.6 Towards a Calculus for Wireless Networks: Recent Advances and What’s Next

*Yuming Jiang (NTNU – Trondheim, NO)*

License  Creative Commons BY 4.0 International license  
© Yuming Jiang

In this talk, a quick recap on the “Wireless Network Calculus” talk at Dagstuhl Seminar 15112 on Network Calculus is first provided. Then, the focus is on introducing some fundamental advances in the topic since that talk. They include the light-tailed property of wireless channel capacity, the dependence structure and copula analysis of wireless channel capacity, and Spatial Network Calculus for wireless networks. The third part is devoted to reliable performance, e.g. 99% packet delivery ratio, in wireless mesh or massive machine-type communications (mMTC). Experimental results from real word IoT wireless networks are used to demonstrate that there is a huge gap. To bridge, an idea is introduced at the end. It is to coordinate transmission among nodes such that the transmission nodes satisfy certain constraints with which the required reliability can be ensured from Spatial Network Calculus analysis.

Main references for further reading:

- F. Sun and Y. Jiang. A Statistical Property of Wireless Channel Capacity: Theory and Application. IFIP Performance 2017
- F. Sun and Y. Jiang. Stochastic Dependence in Wireless Channel Capacity: A Hidden Resource. arXiv 1711.10363 (2017/2019)
- K. Feng and F. Baccelli. Spatial Network Calculus and Performance Guarantees in Wireless Networks. IEEE Trans. Wireless Communications. 23(5): 5033-5047 (2024)

### 3.7 Time Sensitive Networks and Network Calculus

*Jean-Yves Le Boudec (Jouxten-Mézery, CH)*

License  Creative Commons BY 4.0 International license  
© Jean-Yves Le Boudec

Joint work of Jean-Yves Le Boudec, Ahlem Mifdaoui, Ehsan Mohammadpour, Eleni Stai, Ludovic Thomas

Time Sensitive Networks offer guarantees on worst-case delay, worst-case delay variation and zero congestion loss; in addition, they provides mechanisms for packet duplication in order to hide residual losses due to transmission errors. They find applications in many areas such as factory automation, embedded and vehicular networks, audio-visual studio networks, and in the front-hauls of cellular wireless networks. In this talk we describe recent network-calculus results that can be used to analyze time sensitive networks with components such as packet ordering and duplicate removal functions, schedulers, regulators, dampers. We explain why clock non-idealities matter and how to take them into account.

#### References

- 1 Ehsan Mohammadpour, Eleni Stai, Jean-Yves Le Boudec: Improved Network-Calculus Nodal Delay-Bounds in Time-Sensitive Networks. IEEE/ACM Trans. Netw. 31(6): 2902-2917 (2023)
- 2 Ehsan Mohammadpour, Jean-Yves Le Boudec: On Packet Reordering in Time-Sensitive Networks. IEEE/ACM Trans. Netw. 30(3): 1045-1057 (2022)

- 3 Ludovic Thomas, Jean-Yves Le Boudec: On Time Synchronization Issues in Time-Sensitive Networks with Regulators and Nonideal Clocks. *Proc. ACM Meas. Anal. Comput. Syst.* 4(2): 27:1-27:41 (2020)
- 4 Ehsan Mohammadpour, Jean-Yves Le Boudec: Analysis of Dampers in Time-Sensitive Networks With Non-Ideal Clocks. *IEEE/ACM Trans. Netw.* 30(4): 1780-1794 (2022)

### 3.8 Network Calculus Characterization of Congestion Control

*Harvinder Lehal (University of Toronto, CA)*

**License** © Creative Commons BY 4.0 International license  
© Harvinder Lehal

**Joint work of** Harvinder Lehal, Natchanon Luangsomboon, Jörg Liebeherr

**Main reference** Harvinder Lehal, Natchanon Luangsomboon, Jörg Liebeherr: “Network Calculus Characterization of Congestion Control for Time-Varying Traffic”, *CoRR*, Vol. abs/2403.15303, 2024.

**URL** <https://doi.org/10.48550/ARXIV.2403.15303>

Models for the dynamics of congestion control generally involve systems of coupled differential equations which assume that traffic sources saturate the maximum transmissions allowed by the congestion control method. This is not suitable for studying congestion control of intermittent but bursty traffic sources. In this talk, we presented a characterization of congestion control for arbitrary time-varying traffic that applies to rate-based as well as window-based congestion control. We leverage the capability of network calculus to precisely describe the input-output relationship at network elements for arbitrary source traffic and show that our characterization can closely track the dynamics of even complex congestion control algorithms.

### 3.9 Decentralized Reservation Protocols in Time-Sensitive Networking: Applying Network Calculus Without Central Network Overview

*Lisa Maile (Universität Erlangen-Nürnberg, DE)*

**License** © Creative Commons BY 4.0 International license  
© Lisa Maile

**Joint work of** Lisa Maile, Dominik Voitlein, Alexej Grigorjew, Kai-Steffen Jens Hielscher, Reinhard German

**Main reference** Lisa Maile, Dominik Voitlein, Alexej Grigorjew, Kai-Steffen Jens Hielscher, Reinhard German: “On the Validity of Credit-Based Shaper Delay Guarantees in Decentralized Reservation Protocols”, in *Proc. of the 31st International Conference on Real-Time Networks and Systems, RTNS 2023, Dortmund, Germany, June 7-8, 2023*, pp. 108–118, ACM, 2023.

**URL** <https://doi.org/10.1145/3575757.3593644>

Resource reservation is a fundamental mechanism for ensuring quality of service in time-sensitive networks. In the Ethernet technology Time-Sensitive Networking, this can be done decentrally through new resource reservation protocols. For reservation, the standards assume a maximum worst-case latency that is limited at each hop. However, we show that these worst-case latency bounds are not safe. To address this, we propose an alternative to the current standards to allow reservation of time-sensitive traffic with reliable latency guarantees, using Network Calculus. The talk is based on a publication for the Credit-Based Shaper forwarding mechanism, but extends that work to generalise the solution for different forwarding mechanisms and to determine the forwarding parameters, such as the reserved bandwidth.

### 3.10 Statistical Age-of-Information Bounds for Parallel Systems

*Mahsa Noroozi (Leibniz Universität Hannover, DE)*

**License**  Creative Commons BY 4.0 International license  
© Mahsa Noroozi

**Joint work of** Markus Fidler, Jaya Prakash Champati, Joerg Widmer, Mahsa Noroozi  
**Main reference** Markus Fidler, Jaya Prakash Champati, Joerg Widmer, Mahsa Noroozi: “Statistical Age-of-Information Bounds for Parallel Systems: When Do Independent Channels Make a Difference?”, IEEE J. Sel. Areas Inf. Theory, Vol. 4, pp. 591–606, 2023.  
**URL** <https://doi.org/10.1109/JSAIT.2023.3328766>

This work contributes tail bounds of the age-of-information of a general class of parallel systems and explores their potential. Parallel systems arise in relevant cases, such as in multi-band mobile networks, multi-technology wireless access, or multi-path protocols, just to name a few. Typically, control over each communication channel is limited, and random service outages and congestion cause buffering that impairs the age-of-information. The parallel use of independent channels promises a remedy, since outages on one channel may be compensated for by another. Surprisingly, for the well-known case of  $M|M|1$  queues we find the opposite: pooling capacity in one channel performs better than a parallel system with the same total capacity. A generalization is not possible since there are no solutions for other types of parallel queues at hand. In this work, we prove a dual representation of age-of-information in min-plus algebra that connects to queueing models known from the theory of effective bandwidth/capacity and stochastic network calculus. Exploiting these methods, we derive tail bounds of the age-of-information of  $G|G|1$  queues. Tail bounds of the age-of-information of independent parallel queues follow readily. In addition to parallel classical queues, we investigate Markov channels where, depending on the memory of the channel, we show the true advantage of parallel systems. We continue to investigate this new finding and provide insight into when capacity should be pooled in one channel or when independent parallel channels perform better. We complement our analysis with simulation results and evaluate different update policies, scheduling policies, and the use of heterogeneous channels that are most relevant for the latest multi-band networks.

### 3.11 Equivalent versions of Total Flow Analysis and SAIHU tool

*Stéphan Plassart (University Savoie Mont Blanc – Annecy, FR)*

**License**  Creative Commons BY 4.0 International license  
© Stéphan Plassart

**Joint work of** Stéphan Plassart, Jean-Yves Le Boudec, Chun-Tso Tsai, Seyed Mohammadhossein Tabatabaee  
**Main reference** Stéphan Plassart, Jean-Yves Le Boudec: “Equivalent Versions of Total Flow Analysis”, CoRR, Vol. abs/2111.01827, 2021.  
**URL** <https://arxiv.org/abs/2111.01827>  
**Main reference** Chun-Tso Tsai, Seyed Mohammadhossein Tabatabaee, Stéphan Plassart, Jean-Yves Le Boudec: “Saihu: A Common Interface of Worst-Case Delay Analysis Tools for Time-Sensitive Networks”, CoRR, Vol. abs/2303.14565, 2023.  
**URL** <https://doi.org/10.48550/ARXIV.2303.14565>

Total Flow Analysis (TFA) is a method for conducting the worst-case analysis of time sensitive networks without cyclic dependencies. In networks with cyclic dependencies, Fixed-Point TFA introduces artificial cuts, analyses the resulting cycle-free network with TFA, and iterates. If it converges, it does provide valid performance bounds. In the first part of this talk, I show that the choice of the specific cuts used by Fixed-Point TFA does not affect its convergence nor the obtained performance bounds, and that it can be replaced by an alternative algorithm that does not use any cut at all, while still applying to cyclic

dependencies. In the second part, I present Saihu, a common python interface for worst-case delay analysis. Currently it integrates the 3 most used worst-case network analysis tools : xTFA, DiscoDNC, and Panco. Saihu provides a general interface that enables defining the networks in a single XML or JSON file and executing all tools simultaneously without any adjustment for individual tools. Saihu also exports analysis results into formatted reports automatically. Saihu is modular, allowing other worst-case analysis tools to be integrated in the future.

### 3.12 Time-stationary and Event- based Age-of-Information: A Palm calculus bridge

*Amr Rizk (Universität Duisburg-Essen, DE)*

**License** © Creative Commons BY 4.0 International license  
© Amr Rizk

**Joint work of** Amr Rizk, Jean-Yves Le Boudec

**Main reference** Amr Rizk, Jean-Yves Le Boudec: “A Palm Calculus Approach to the Distribution of the Age of Information”, IEEE Trans. Inf. Theory, Vol. 69(12), pp. 8097–8110, 2023.

**URL** <https://doi.org/10.1109/TIT.2023.3326381>

A key metric to express the timeliness of status updates in latency-sensitive networked systems is the age of information (AoI), i.e., the time elapsed since the generation of the last received informative status message. State-of-the-art approaches to analyzing the AoI rely on queueing models that are composed of one or many queueing systems endowed with service order, e.g., FIFO, LIFO, or last-generated-first-out order. A major difficulty arising in these analysis methods is capturing the AoI under message reordering when the delivery is non-preemptive and non-FIFO, i.e., when messages can overtake each other and the reception of informative messages may obsolete some messages that are underway.

This talk which is based on our work [1] describes the derivation of computable exact formulations for the distribution of AoI in non-preemptive, non-FIFO systems where the main ingredients of our analysis are Palm calculus and time inversion.

#### References

- 1 Amr Rizk and Jean-Yves Le Boudec. *Palm Calculus Approach to the Distribution of the Age of Information*. In IEEE Transactions on Information Theory, vol. 69, no. 12, pp. 8097-8110, 2023, doi: 10.1109/TIT.2023.3326381.

### 3.13 Logic-Based Formal Analysis of Network Performance

*Mina Tahmasbi Arashloo (University of Waterloo, CA)*

**License** © Creative Commons BY 4.0 International license  
© Mina Tahmasbi Arashloo

In recent years, there has been a trend toward using programs written in high-level domain-specific languages to specify network packet processing behavior. This has opened up possibilities to use program analysis techniques to formally reason about network properties such as reachability and absence of forwarding loops. Most of the existing work in this area focuses on reasoning about the functional correctness of computer networks. In this talk, we present our recent efforts in using logic-based formal analysis techniques to reason about network performance properties such as throughput, latency, starvation, and fairness, and its

potential synergies with network calculus. In particular, we discuss (1) how using logic-based formal analysis can help relax some of the assumptions and simplifications one may need to make in network calculus about modeling the arrival pattern of traffic, the packet processing logic, and packet loss, and (2) how network-calculus-like abstractions can help reduce the significant computational overhead that logic-based formal analysis of performance properties is prone to in comparisons with functional-correctness properties.

### 3.14 Factors Limiting the Modularity of xTFA

*Ludovic Thomas (CNRS – Villers-lès-Nancy, FR)*

License  Creative Commons BY 4.0 International license  
© Ludovic Thomas

Joint work of Ludovic Thomas, Ahlem Mifdaoui, Jean-Yves Le Boudec

Experimental modular Total-Flow Analysis (xTFA) is an open-source experimental Python tool for computing end-to-end latencies in time-sensitive networks. Through a modular conception based on computational blocks and flow-state partitions, xTFA supports a wide variety of models, including redundancy functions, interleaved regulators, cyclic dependencies, and clock non-idealities. But the modularity of the tool is limited because of the assumptions that the different models make and the variety of these assumptions. This talk discusses the origin of these limitations and proposes open question towards a classification of assumptions for service-curves properties in time-sensitive networks.

### 3.15 Performance and Scaling of Parallel Systems with Blocking Start and/or Departure Barriers

*Brenton Walker (Leibniz Universität Hannover, DE)*

License  Creative Commons BY 4.0 International license  
© Brenton Walker

Parallel systems divide jobs into smaller tasks that can be serviced by many workers at the same time. Some parallel systems have blocking barriers that require all of their tasks to start and/or depart in unison. This is true of many parallel Machine Learning workloads, and popular Apache Spark engine has, for this reason, recently added support for Barrier Execution Mode (BEM).

The drawback of these barriers is reduced performance and stability, compared to equivalent non-blocking systems. We derived analytical expressions for the stability for BEM systems and extend results from Network Calculus to derive waiting and sojourn time bounds.

Our results show that for a given utilization ( $\rho$ ) and number of workers ( $s$ ), there is an optimal degree of parallelism ( $k$ ) that balances waiting time and execution time to minimize sojourn times. This complements prior work on so called "tiny tasks", where  $\frac{k}{s} \gg 1$ , to show that among systems with barriers, the unstable Split-Merge model ( $k = s$ ) is the anomaly, and taking  $\frac{k}{s} < 1$  or  $\frac{k}{s} > 1$  is actually much better.

### 3.16 Improving algorithms and software for DNC

Raffaele Zippo (*University of Pisa, IT*)

**License** © Creative Commons BY 4.0 International license  
© Raffaele Zippo

**Joint work of** Raffaele Zippo, Paul Nikolaus, Giovanni Stea, Andrea Trasacco

**Main reference** Raffaele Zippo, Giovanni Stea: “Nancy: An efficient parallel Network Calculus library”, *SoftwareX*, Vol. 19, p. 101178, 2022.

**URL** <https://doi.org/10.1016/J.SOFTX.2022.101178>

**Main reference** Raffaele Zippo, Paul Nikolaus, Giovanni Stea: “Extending the network calculus algorithmic toolbox for ultimately pseudo-periodic functions: pseudo-inverse and composition”, *Discret. Event Dyn. Syst.*, Vol. 33(3), pp. 181–219, 2023.

**URL** <https://doi.org/10.1007/S10626-022-00373-5>

**Main reference** Raffaele Zippo, Paul Nikolaus, Giovanni Stea: “Isospeed: Improving (min, +) Convolution by Exploiting (min, +)/(max, +) Isomorphism”, in *Proc. of the 35th Euromicro Conference on Real-Time Systems, ECRTS 2023, July 11-14, 2023, Vienna, Austria, LIPICs, Vol. 262*, pp. 12:1–12:24, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

**URL** <https://doi.org/10.4230/LIPICs.ECRTS.2023.12>

In this talk, I present results and work in progress improving the existing software and algorithms for DNC computations, particularly those built on the Nancy computational library. First, I discuss the optimizations stemming from the duality between (min,+) and (max,+), which we call isospeed algorithms. Then, I present some work in progress that may be useful to the community: Nancy HTTP wrappers, to use the library from other languages; Nancy.Interactive to improve the testing and teaching workflow; Nancy.Expressions, opening up new ways to optimize computations transparently to the user; and DNC Benchmarking, a project aimed at benchmarking DNC implementations to highlight the impact of optimizations on runtime. Finally, a note on the need for practical examples to highlight, especially to those outside our community, where the hard computations are in DNC, and thus improve the chances of publication of this kind of results.

## 4 Working groups

### 4.1 Network Calculus and Machine Learning/Artificial Intelligence

Michael Beck (*University of Winnipeg, CA*)

**License** © Creative Commons BY 4.0 International license  
© Michael Beck

Our group discussed three areas in which machine learning and network calculus overlap, all of them related to hard optimization problems. These are i) optimal transformation of flows to allow the analysis of feedforward networks with deterministic network calculus (DNC); ii) optimal cutting of flows to break cyclic dependencies in arbitrary networks; iii) determining the schedule in IEEE 802.1Qbv time-aware schedulers.

#### 4.1.1 Optimal topology manipulation in feedforward networks

To apply the benefits of a Pay-Multiplexing-Only-Once (PMOO) analysis in DNC the corresponding network topology must be a nested feed-forward network. This means that the service elements in the network can be ordered  $1, \dots, n$  such that i) for each flow all nodes it visits are traversed in increasing order and ii) for any two flows  $F_1, F_2$  we have either

$$s_{F_1} > e_{F_2} \text{ OR } e_{F_1} < s_{F_2} \quad (1)$$

or

$$\mathcal{N}(F_1) \subseteq \mathcal{N}(F_2) \text{ or } \mathcal{N}(F_1) \supseteq \mathcal{N}(F_2), \quad (2)$$

where  $s_F$ ,  $e_F$ , and  $\mathcal{N}(F)$  denote the first, last, and all nodes a flow  $F$  traverses respectively.

In networks that have interleaving flows instead, i.e. are not nested, the PMOO analysis is not directly applicable. Since PMOO leads to generally better bounds, it is thus desirable to modify the topology, this is done by “cutting” one or more flows. For example, consider a three node network with the two flows  $\mathcal{N}(F_1) = \{1, 2\}$  and  $\mathcal{N}(F_2) = \{2, 3\}$ . In this case it is possible to consider flow  $F_1$  leaving the network after traversal of node 1 and re-entering it at node 2, i.e., an output bound of flow  $F_1$  is computed after being served at node 1 and this output bound serves as a new arrival bound for a flow  $F'_1$  with  $\mathcal{N}(F'_1) = \{2\}$ . Alternatively, flow  $F_2$  could be cut after traversing node 2. The resulting performance bounds are generally not the same leaving a decision problem for the practitioner, which does not have an obvious analytical solution even for modest network sizes.

An alternative to cutting flows is to prolong flows to create a nested topology. In the above example we could also extend either of the two flows to traverse all three nodes leading to two more possible topologies, which are nested and serve as an upper bound for the original network with respect to calculable performance guarantees. In more complex networks several steps of flow prolongations and flow cuttings will have to be performed to reach a nested topology.

Finding an optimal sequence for these steps in general feed-forward networks is a problem of high complexity. To find such a sequence a machine learning approach can be applied, which is based on graph neural networks (GNN). For this methodology to be applied the network must be represented as a graph which represents possible cuts of flows (a cutting graph) or prolongation of flows (a prolongation graph). Creating a single graph that captures both, cuts and prolongations, has – to the authors knowledge – not been reported, yet.

Prolongation and cutting graphs are very similar to each other, thus for brevity we describe here only cutting graphs. They consists of four types of nodes: i) server nodes, ii) flow nodes, iii) ordering nodes, and iv) decision nodes. Together with edges and node labels the cutting graph is a one-to-one mapping of a network. The cutting graph is created as follows:

- Each server of the network is represented as a server node with a label that represents the server’s service guarantee. The server nodes are connected via edges in tandem according to their natural ordering  $1, \dots, n$  (see the first condition of feedforward topologies at the beginning of this subsection).
- Each flow of the network is represented as a flow node with a label that represents the flow’s (upper) arrival guarantee.
- For each flow a number of ordering nodes, equal to the number of servers the flow traverses, is added to the graph. The ordering nodes are labeled by the hop count of the flow at each server and are connected via edges to the flow node and the respective server node.
- For each flow and each consecutive pair of ordering nodes a decision node is added to the graph and connected with three edges to the pair of ordering nodes and the respective flow node. These nodes are initially unlabeled.

For a visualization of a network and its corresponding cutting graph see [1].

Having such a cutting graph at hand a GNN can be used to determine labels to the ordering nodes. These labels will be binary and represent the decision to perform the cut or not. To train the GNN a supervised learning approach has been followed in [1] that requires training data for which the optimal solution is already known. Due to the complexity of



the problem the such generated training data only contains graphs of relatively small size for which an optimal solution can still be obtained. Our group discussed the idea whether a model trained on such data can generalize successfully to networks of larger size. This research question holds for both cutting graphs and prolongation graphs.

Since, optimal solutions for larger topologies are out of reach the problem could also be approached by unsupervised machine learning methods instead. The intermediate obtained performance bound will serve as feedback to the unsupervised methodology, for example, as a fitness function in a genetic algorithm.

#### 4.1.2 Optimal cutting in cyclic networks

Very related to the above is the question of which cuts to perform to break cyclic flow dependencies in non-feedforward networks. In such a network there exists at least one cycle of servers which can be traversed just by following the direction of one or more flows. Such cycles prevent the direct application of the PMOO analysis. Which flows to cut at which positions to achieve optimal performance guarantees is not obvious. The application of supervised or unsupervised machine learning methods similar to those described above should be investigated.

#### 4.1.3 Time-Aware Scheduling in Time Sensitive Networks

The last topic discussed by our group was how to determine an optimal schedule for IEEE 802.1Qbv schedulers. More precisely, given

- an overall service description for the service element,
- arrival curves for each flow (each flow enters one of the 8 queues, with more than one flow per queue possible),
- the length of the cyclic schedule, and
- prioritization between queues,

how should the queue's gates be opened and closed to meet the delay requirements for each of the 8 traffic classes. To note here is, that the opposite problem of deriving delay bounds from a given schedule is relatively easy.


Since the schedule can be arbitrarily structured we face another hard optimization problem. We see an opportunity to deploy machine learning methods for synthesizing a schedule according to the delay objectives. An unsupervised methodology seems appropriate here, with a feedback that is based on the margins by which the delay requirements are (not) met. Due to the continuous introduction and ceasing of flows the gate schedules must be created online, which in turn, puts a limit on the complexity of the models that can be used.

#### References

- 1 Fabien Geyer, Steffen Bondorf, *Graph-based deep learning for fast and tight network calculus analyses*, IEEE Transactions on Network Science and Engineering, 8, 1, p. 75-88. 2020

## 4.2 Network Calculus versus other types of real-time network analysis

Anne Bouillard (*Huawei Technologies – Boulogne-Billancourt, FR*) and Jörg Liebeherr (*University of Toronto, CA*)

License  Creative Commons BY 4.0 International license  
 © Anne Bouillard and Jörg Liebeherr

Recent studies have highlighted similarities between different analyses approaches for real-time systems:

- The equivalence between Real-Time Calculus (RTC) and the variable capacity node in Network Calculus (NC) was established over a decade ago.
- More recent research identifies similarities with Response Time Analysis (RTA) and even stream models.

Despite these similarities, significant differences in the approaches remain. Investigating the potential of NC formalism could be beneficial.

### 4.2.1 Comparative Analysis: Other Methods vs. Network Calculus

- NC does not support multicore analysis.
- Packet losses are seldom considered in NC (there are only a few works about it)
- NC lacks Directed Acyclic Graph (DAG) tasks to represent task patterns.
- NC also does not consider behaviors depending on a state, that could be represented by an automaton

### 4.2.2 Focus on RTC and NC Comparison

The similarities have been established quite long ago, since RTC is a variation of NC and has been defined to be applied to real-time systems, whereas NC was previously targeting communication systems. The primary difference lies in the time domain (non-negative reals in NC vs. the whole real number in RTC), and this makes a difference, especially when considering lower arrival curves (which is doable, but not often done in NC).

Another difference in the approach rather than in the theory is that NC will abstract the model by simpler curves (a few token-buckets), here RTC will consider stair-case curves (based on the packet level).

Some generalization have also been considered, like interleaved of packets of different type and differentiated service among them.

We also pointed a difference in the approach: while NC is analysis a network when the scheduling parameters are fixed (allowing modular or global approaches), the RTC is modular and in order not to be too pessimistic in the composition, it chooses the scheduler.

### 4.2.3 A Common Open Issue: Back Pressure

Back-pressure model analysis remains unresolved. Some have looked at the model (for NoC or wormhole routing analysis), and failed to have an accurate analysis. Further theoretical developments may provide solutions.

### 4.2.4 Takeaway

It might be an opportune time to draft a review paper on *20 Years of RTC*.

#### 4.2.5 List of Acronyms:

- DAG: Directed Acyclic Graph
- NC: Network Calculus
- NoC: Network on Chip
- RTA: Response Time Analysis
- RTC: Real-Time Calculus

### 4.3 Core of the Network Calculus Theory

*Marc Boyer (ONERA – Toulouse, FR)*

License  Creative Commons BY 4.0 International license  
© Marc Boyer

The group discussed different versions of service, which offered different points of view (Adaptive service, Real-Time Calculus – RTC, strict service, etc.).

Some discussions also on the equivalence between  $(\min,+)$  and  $(\max,+)$  dioids. Rather than using them one at a time, one can think about  $(\min\text{-plus})$  and  $(\max\text{-plus})$  at the same time. Starting points can be found in the 1992 book by Baccelli/Cohen/Olsder/Quadrat [1]. Since convolution is a Minkowski sum, an exploration of algebras of Minkowski systems may lead to progress. The Moreau conjugacy a generalisation of legendre transform, has also been mentioned.

We also speak about the question of rare perturbation. The deterministic network calculus is designed to bound the worst case, but not to quantify how rare/frequent are this worst cases. Considering a sequence of message, all having a delay of 1ms except one out of 1000 that have a delay of 10ms. Then network calculus is not able to make the difference with a sequence with all delays being 10ms. And this is not related to the bounding methods, even the definition of delay is not able to capture it.


Additional, short discussion have been done on how to model real systems with notion of mode and mode change and on interesting non-linear systems.

#### References

- 1 F. Baccelli, G. Cohen, G.J. Olsder, J.P. Quadrat. *Synchronization and Linearity: An algebra for discrete event systems*. ISBN: 047193609X. John Wiley & Sons Ltd, 1992

### 4.4 Network Calculus Tools

*Lisa Maile (Universität Erlangen-Nürnberg, DE) and Anja Hamscher (RPTU – Kaiserslautern, DE)*

License  Creative Commons BY 4.0 International license  
© Lisa Maile and Anja Hamscher

#### 4.4.1 Introduction

This section outlines the discussion held on NC tools and their functionalities, mainly “Nancy” and “Saihu”.

#### 4.4.2 New “Nancy” Functionality

Nancy [2] is a versatile C# library specifically designed for Network Calculus computations. It implements min-plus and max-plus operators and can manage ultimately pseudo-periodic piecewise affine curves. Nancy is released under the MIT license and developed by Raffaele Zippo and Giovanni Stea from the University of Pisa, Italy.

The new functionality of Nancy includes the ability to embed Nancy in other languages through HTTP+JSON. During the discussion, a live demonstration showcased how this integration works. Feedback on the current implementation addressed several points:

- Compatibility is not limited to Windows.
- Integration into existing projects is a key motivation for this development.
- Nancy utilizes Plotly for plotting, which can be reimplemented as needed.
- The need for garbage collection and storage management for large networks was highlighted as an area requiring further investigation.

#### 4.4.3 Presentation of “Saihu”

Saihu [1] is a Python interface that integrates several major worst-case network analysis tools, including the tools xTFA (supports TFA), DiscoDNC (supports LUDB, PMOO, SFA), and Panco (supports PLP and ELP). These tools are frequently used in time-sensitive network analysis. Saihu provides a general interface that allows users to define networks in a single XML or JSON file and execute all tools simultaneously without needing to adjust individual tools.

Additionally, Saihu exports analysis results into formatted reports automatically and offers automatic network generation significantly reducing the required work of users. The modular nature of the package allows for the incorporation of more tools in the future. During the discussion, the following points were emphasized:

- Extension for new tools can be achieved using a common JSON format.
- The syntax of JSON files needs to be compatible with various tools to ensure smooth integration.

#### 4.4.4 Further Aspects

The user experience of the tools was discussed, focusing on ease of use, integration capabilities, and the overall satisfaction of users with the functionalities provided by both Nancy and Saihu.

A comprehensive list of tools is available on Wikipedia, which serves as a valuable resource for users looking to explore different tools available for network analysis and other related tasks.

One of the issues discussed was the challenge of allocating time and resources for tool development in an academic setting. The results of programming efforts alone are often difficult to publish, highlighting a significant hurdle for researchers and developers in academia.

The development of a benchmark was proposed as a beneficial step to compare different tools effectively. Such benchmarks would provide standardized criteria for evaluating tool performance and capabilities, facilitating better decision-making for users and developers.

#### 4.4.5 Conclusion

The discussion provided valuable insights into the functionalities and future directions for tools like Nancy and Saihu. Addressing user feedback, compatibility, and resource challenges will be crucial for the continued development and adoption of these tools in various applications.

**References**

- 1 Tsai, Chun-Tso et al. “Saihu: A Common Interface of Worst-Case Delay Analysis Tools for Time-Sensitive Networks.” ArXiv abs/2303.14565 (2023).
- 2 R. Zippo and G. Stea, “Nancy: An efficient parallel network calculus library”, SoftwareX, vol. 19, Jul. 2022.

## Participants

- Michael Beck  
University of Winnipeg, CA
- Steffen Bondorf  
Ruhr-Universität Bochum, DE
- Anne Bouillard  
Huawei Technologies –  
Boulogne-Billancourt, FR
- Marc Boyer  
ONERA – Toulouse, FR
- Peter Buchholz  
TU Dortmund, DE
- Almut Burchard  
University of Toronto, CA
- Georg Carle  
TU München – Garching, DE
- Samarjit Chakraborty  
University of North Carolina at  
Chapel Hill, US
- Vlad-Cristian Constantin  
RPTU – Kaiserslautern, DE
- Hugo Daigmorte  
RealTime-at-Work – Nancy, FR
- Markus Fidler  
Leibniz Universität  
Hannover, DE
- Anaïs Finzi  
TTTech Computertechnik –  
Wien, AT
- Stéphane Gaubert  
INRIA & CMAP, Ecole  
polytechnique – Palaiseau, FR
- Damien Guidolin-Pina  
RealTime-at-Work – Nancy, FR
- Anja Hamscher  
RPTU – Kaiserslautern, DE
- Max Helm  
TU München – Garching, DE
- Kai-Steffen Jens Hielscher  
Universität Erlangen-  
Nürnberg, DE
- Yuming Jiang  
NTNU – Trondheim, NO
- Kai Lampka  
NXP Semiconductors –  
München, DE
- Jean-Yves Le Boudec  
Juxtens-Mézery, CH
- Harvinder Lehal  
University of Toronto, CA
- Jörg Liebeherr  
University of Toronto, CA
- Lisa Maile  
Universität Erlangen-  
Nürnberg, DE
- Mahsa Noroozi  
Leibniz Universität  
Hannover, DE
- Xi Peng  
Huawei Technologies –  
Hong Kong, HK
- Stéphan Plassart  
University Savoie Mont Blanc –  
Annecy, FR
- Amr Rizk  
Universität Duisburg-Essen, DE
- Giovanni Stea  
University of Pisa, IT
- Mina Tahmasbi Arashloo  
University of Waterloo, CA
- Ludovic Thomas  
CNRS – Villers-lès-Nancy, FR
- Brenton Walker  
Leibniz Universität  
Hannover, DE
- Kui Wu  
University of Victoria, CA
- Raffaele Zippo  
University of Pisa, IT

