42° Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2024

Tutorial Proposal

[1]

Abstract. *Programmable Network Interface Cards (SmartNICs) have gained increasing momentum due to their flexibility to offload complex networking tasks from the host CPU to a programmable hardware architecture. Despite the performance gains (e.g., lower latency), programming, debugging, and operating SmartNICs pose challenges due to diverse hardware architectures (e.g., System-on-Chip, FPGA, and ASIC), various programming languages, and operation modes. This tutorial aims to shed light on the design principles and operation of modern SmartNICs, covering hardware architectures, programming software ecosystem, performance capabilities and open research challenges in this evolving domain, The tutorial concludes with a hands-on experience involving cutting-edge SmartNICs such as Nvidia BlueField, Xilinx Alveo, and Netronome NFP.*

1. Identification data

1.1. Title

SmartNICs: The Next Leap in Networking

1.2. Authors

Marcelo Caggiani Luizelli
Universidade Federal de Pampa (Unipampa)
Av. Tiaraju, 810, Alegrete/RS
marceloluizelli@unipampa.edu.br

Francisco Vogt
Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
f234632@dac.unicamp.br

Guilherme Mendes Vieira de Matos
Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar)
Rodovia João Leme dos Santos, km 110, Sorocaba/SP

guilherme.matos@estudante.ufscar.br

Weverton Luis da Costa Cordeiro
Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS)
weverton.cordeiro@inf.ufrgs.br

Alberto Egon Schaeffer Filho
Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS)
alberto@inf.ufrgs.br

Marcos Schwarz
Diretoria de Pesquisa, Desenvolvimento e Inovação (DPDI)
Rede Nacional de Ensino e Pesquisa (RNP)
marcos.schwarz@rnp.br

Fábio Luciano Verdi
Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar)
Rodovia João Leme dos Santos, km 110, Sorocaba/SP
verdi@ufscar.br

Christian Esteve Rothenberg
Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
chesteve@dca.fee.unicamp.br

1.3. Indication of the authors who will present the tutorial

Marcelo Caggiani Luizelli
Marcos Schwarz
Francisco Germano Vogt
Guilherme Mendes Vieira de Matos

2. General data

2.1. Main goal

The ability to program the network data plane has reshaped the network operations and management landscape, opening up a multitude of opportunities for delivering custom-made networking solutions [Kianpisheh and Taleb 2023]. By running home-brewed networking solutions within network programmable devices, network operators and practitioners have an opportunity to close the control-management loop and make per-packet forwarding decisions at line rate, on the order of nanoseconds. Recently, there has been

an increasing interest from both academic and industrial sectors towards programmable NICs (Network Interface Cards), commonly referred to as SmartNICs. These devices are gaining attention due to their ability to efficiently offload a wide range of intricate networking tasks, as well as tasks unrelated to networking, offering enhanced flexibility in the process. Examples include in-network caching [He et al. 2023] and in-network machine learning [Swamy et al. 2022] – just to name few examples. Nevertheless, programming, debugging, and operating SmartNICs remain challenging. This difficulty primarily stems from the diverse hardware architectures they rely on, such as System-on-Chip, FPGA, and ASIC, coupled with varying programming languages and operation modes [Kianpisheh and Taleb 2023]. To shed light on the design of emerging in-network solutions, this tutorial aims to introduce essential principles related to the design and operation of state-of-the-art SmartNICs. We aim to cover and discuss different hardware architectures, and the available programming software ecosystem. We will dive into the performance limitations of existing SmartNICs, and discuss tailored networking applications for them. In the end, we will conduct a hands-on experience with state-of-the-art SmartNICs (Nvidia BlueField, Xilinx Alveo, and Netronome NFP).

2.2. Target audience profile

This tutorial is aimed at undergraduate and postgraduate students, spanning master's and doctoral levels, as well as professionals with a keen interest in programmable networks. Prerequisite knowledge includes a solid understanding of computer network concepts and basic programming.

3. Tutorial structure

This tutorial will be organized according to the sections below:

1. Introduction and Motivation (3 pages)
2. Fundamentals of SmartNICs (10 pages)
3. Software Ecosystem (4 pages)
4. Performance Benchmarking (4 pages)
5. Commercial Products & Market Perspectives (2 pages)
6. Deep Diving into Real SmartNIC Usecases: a hands-on approach (15 pages)
7. Challenges and Future Trends (4 pages)
8. Closing Remarks (2 pages)

This structure ensures a logical and systematic progression through the essential aspects covered in the tutorial, providing a clear roadmap for the audience.

4. Summary of the content to be covered

I - Introduction and motivation (3 pages)

In this section, we will explore the evolving concept of SmartNICs, delving into a historical timeline of modern Network Interface Cards (NICs). The discussion will include motivating use cases, as well as the performance requirements for upcoming applications, such as multi-sensory extended reality, brain-computer interfaces, and haptic interaction.

These specialized network interface cards, equipped with programmable capabilities, address the evolving challenges posed by modern data-intensive applications and

complex network environments (i.e., higher throughput/lower latency). SmartNICs play a pivotal role in offloading and accelerating a multitude of tasks, ranging from networking functions to non-networking workloads [Kianpisheh and Taleb 2023]. Furthermore, their ability to process packets at the network edge can enhance overall system performance, reduce latency, and optimize resource utilization. As the demand for flexible and efficient network solutions continues to grow, SmartNICs emerge as a critical component, offering a scalable and adaptable approach to meet the dynamic requirements of networking.

The evolution of NICs over the decades reflects the dynamic progression of networking technologies to meet the ever-increasing demand of computing systems. In the 1980s, NICs emerged as simple devices to enable computer connectivity to Ethernet networks, primarily utilizing coaxial cables and 10 Mbps Ethernet technologies. In the 1990s, NICs were adapted to support faster Ethernet speeds, reaching up to 1 Gbps, and wireless NICs paved the way for mobile networks. In turn, it was in the 2000s that the concept of offloading specific networking functions (e.g., TCP offloading such as Generic Receive Offloading) to dedicated hardware marked a shift in network interface development. With the surge of data center requirements, other features were designed/offloaded to NICs (e.g, SR-IOV). The 2010s was characterized by the growth of cloud computing, which established higher networking transmission rates such as 10-40 Gbps to address data center needs. Nowadays, NICs (or now SmartNICs) have evolved to support network speeds that go beyond 100 Gbps, and incorporate programmable units (e.g., FPGA, CPU/NPU) into the NIC pipeline. As computing environments faced escalating demands from data-intensive applications and intricate network architectures, the need for programmability and adaptability in network interfaces became increasingly evident. Advancements in programmable hardware, coupled with the paradigm shift toward Software-Defined Networking (SDN), further propelled the evolution of SmartNIC technology.

At the end of this section, the reader will be motivated to deep dive into existing SmartNIC technologies and the existing programming toolchains.

II - Fundamentals of SmartNICs (10 pages)

This section will provide a solid background on SmartNICs and on the Portable NIC Architecture (PNA) [(PNA) 2023]. We will start with a broad overview of the network packet processing pipeline in existing NICs. Then, we will cover the state-of-the-art SmartNIC architectures: (i) Nvidia BlueField, (ii) Xilinx Alveo, and (iii) Netronome NFP-4000.

As previously mentioned, NICs have evolved to support network speeds that go beyond 100 Gbps, while incorporating programmable units. The hardware architecture necessary for achieving high-speed network packet processing requires a high degree of parallelism to achieve performance scalability in NIC programs. To address this, current programmable NICs rely on multiple hardware architectures including: (i) ASIC (e.g., Netronome NFP); (ii) System-on-Chip (e.g., NVIDIA BlueField); and (iii) FPGA (e.g., Xilinx Alveo). Our discussion in this section will focus on the refereed architectures, which play a central role in many scientific studies [Liu et al. 2019, Min et al. 2021, Schuh et al. 2021, Wei et al. 2023].

SmartNICs typically employ an alternative processing approach in comparison

to an ASIC switch (e.g., Tofino), wherein a packet is directed to a specific processing engine following a run-to-completion model. For example, Nvidia BlueField adopts a "disaggregated RMT" architecture [Chole et al. 2017]. In this design, a group of ASIC packet engines handles header computation and retrieves match/action (MA) entries from SRAM via a memory bus. In contrast, Netronome Agilio utilizes a collection of SoC-based CPU cores for packet processing, with corresponding entries situated in a more distant memory hierarchy [Xing et al. 2023]. In the case of multicore SmartNICs, packet latency can vary based on the program structure, including factors such as the number of Match/Action tables and their match types. Additionally, packets following distinct execution paths within the same program may encounter varying levels of latency.

At the end of this section, the reader will have a clear understanding of the hardware architectural elements that impact the performance of SmartNIC applications. Also, they will be familiar with different programming models that fit the architectural characteristics of each SmartNIC.

III - Software Ecosystem (4 pages)

In this section, we will present the existing software platforms to program, debug, deploy, and manage applications into SmartNICs.

Although the P4 language has become the *de facto* language for data plane programming, we have observed that programming SmartNICs still requires a substantial effort from the research community to provide open programming standards and code interoperability. Most of the SmartNIC vendors provide a P4 compiler (or transpiler). However, we have witnessed that depending on the program logic we want to implement, we are compelled to use P4 "externs" to fully use the computation power available in SmartNICs. Therefore, the P4 code is usually used to program the "basic" pipeline behavior (i.e., parser/deparser, header definitions, match&action tables), while more CPU (or memory) intensive tasks are left to P4 externs. As an example, the Netronome provides a Micro-C as an alternative to write externs, while NVIDIA introduced the DOCA platform which extends the DPDK library in C language.

Apart from the existing differences in writing SmartNIC code to different hardware platforms, each vendor has its way to compile, debug, and run the code in the target. At the end of this section, the reader will have a broader view of what exists and what can be used for each hardware platform.

IV – Performance Benchmarking (4 pages)

When designing applications for SmartNICs, it is well-known that achieving optimal performance depends on complex tuning [Viegas et al. 2021, Xing et al. 2023]. Traditionally, this tuning is carried out in a target-specific manner by the SmartNIC compiler or developer, employing diverse low-level programming optimizations such as C optimizations for Nvidia BlueField and Micro-C for Netronome Agilio.

While P4 ASIC switch compilers guarantee performance and packet processing operation at lines rate – as long as the P4 code fits into the hardware resources – SmartNICs usually do not have hard "constraints" (e.g., number of pipeline stages) and, therefore, require a more careful tuning process. Current SmartNIC architectures, as exemplified by [Netronome 2020], typically do not impose constraints on the number of opera-

tions executed by the data plane within a single pipeline stage. For example, a SmartNIC application might perform an unrestricted number of reads and writes to registers in a specific stage of the packet processing pipeline. Additionally, the same application could recirculate the ingress packet multiple times to simulate a loop-based mechanism. These instances represent basic operations commonly employed by more complex applications, such as those used in in-network clustering [Xiong and Zilberman 2019]. In an experimental evaluation using the Netronome SmartNIC, Vegas et al. [Viegas et al. 2021] have shown that the line-rate throughput is bounded by (i) the number of register operations (up to 10 operations), (ii) the number of multiple match+action tables used in the pipeline (up to 5), and (iii) the number of cryptography operations (up to 10).

At the end of this section, the reader will understand the performance limitations of different SmartNICs to effectively design efficient data plane applications.

V - Commercial Products & Market Perspectives (2 pages)

In this section, we will discuss the SmartNIC market perspective. We will cover the market share, adoption of existing SmartNICs, CAPEX/OPEX, and potential commercial opportunities in this domain.

VI - Deep Diving into Real SmartNIC Use Cases: a Hands-on Approach (15 pages)

In this section, we will explore a range of SmartNIC use cases from a practical perspective. We will cover three uses cases, one for each hardware platform, and, in all of them, we will describe the entire SmartNIC programming process: coding, compiling, debugging, deploying, and evaluating.

We expect to conduct a hands-on activity with at least one of the existing platforms. We aim to provide remote access to servers and SmartNICs resources available in the research groups in which the proponents of this proposal work (i.e., UNICAMP, UNIPAMPA, UFRGS, UFSCAR, and RNP). Alternatively, depending on the availability and logistics, we may bring a few servers to the conference. The other use cases will be shown by the instructors either through remote access to servers or by pre-recorded videos.

In all hands-on use cases, we will provide offline resources (via git repositories) so that this tutorial can become a reference to other SmartNIC programmers in the future.

1. **Netronome NFP4000:** we will showcase a simple forwarding mechanism (e.g., IPv4) implemented in P4 and deployed in the Netronome SmartNIC. The idea is to provide fundamental insights on the whole process, from P4 coding to deployment and testing on the NIC. Additionally, we will also cover configuration aspects, SRIOV (Single Root I/O Virtualization), instance control, and control plane setup. This expansion will offer a more comprehensive and detailed understanding of the environment and operations involved;
2. **Nvidia BlueField-2:** we will demonstrate the design and implementation of a C language-based code written using the NVIDIA DOCA platform. The running example aims to describe the basic functionalities of DOCA Flow module in the BlueField-2 architecture. We will also dive into deployment features and testing. The idea is to provide fundamental insights on the whole process, from C coding to deployment, and testing.

3. **Xilinx Alveo SN-1000:** we will showcase the design and operation of a P4 code following the Xilinx FPGA hardware architecture. We will explore the Xilinx VitisP4 development framework to show simple P4 code prototyping, as well as the design of FPGA externs.
4. **Nvidia Connectx-5/6:** we will demonstrate offload capabilities using different frameworks (i.e. DPDK, RDMA, switchdev) that can be used for virtual switch (i.e. Open vSwitch) and virtual router (i.e. VPP) hardware accelerations. We will use resources from existing testbeds (i.e. NSF FABRIC, RNP Testbed Service) to allow the replication of the experiments.

VII - Challenges and Future Trends (4 pages)

In this section, we will discuss existing challenges and future research trends in the area of SmartNICs, including: (i) the absence of standardized P4/data plane specifications, (ii) limitations in troubleshooting; (iii) life-cycle orchestration of SmartNICs; (iv) higher costs in comparison to traditional NICs; (v) potential delays in the implementation due to SmartNIC complexity and constrained programmability. These considerations highlight the difficulties and trade-offs associated with incorporating SmartNICs into the design of network architectures.

VIII - Closing Remarks (2 page)

This section will conclude the tutorial by summarizing the key takeaways learned in the process of programming SmartNICs, and discuss potential research opportunities within this domain.

As the demand for high-speed data processing continues to grow exponentially, SmartNIC technology has paved the way for significant advancements in networking and data processing for AI training and cloud computing. These specialized network interface cards have proven to be instrumental in overcoming the challenges of modern computing, enabling efficient data offloading, acceleration of critical tasks, and seamless integration with the existing infrastructure. With ongoing research and development, we can expect further innovations in SmartNIC technology, unlocking new possibilities and driving the next wave of advancements in networking and data-driven applications.

Expected total pages: 40 - 45 pages

5. Main bibliography used in tutorial

References

Chole, S., Fingerhut, A., Ma, S., Sivaraman, A., Vargaftik, S., Berger, A., Mendelson, G., Alizadeh, M., Chuang, S.-T., Keslassy, I., Orda, A., and Edsall, T. (2017). Drmt: Disaggregated programmable switching. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, page 1–14, New York, NY, USA.

He, Y., Wu, W., Le, Y., Liu, M., and Lao, C. (2023). A generic service to provide in-network aggregation for key-value streams. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS 2023, page 33–47, New York, NY, USA.

Kianpisheh, S. and Taleb, T. (2023). A survey on in-network computing: Programmable data plane and technology specific applications. *IEEE Communications Surveys Tutorials*, 25(1):701–761.

Liu, M., Cui, T., Schuh, H., Krishnamurthy, A., Peter, S., and Gupta, K. (2019). Offloading distributed applications onto smartnics using ipipe. In *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM '19, page 318–333, New York, NY, USA.

Min, J., Liu, M., Chugh, T., Zhao, C., Wei, A., Doh, I. H., and Krishnamurthy, A. (2021). Gimbal: Enabling multi-tenant storage disaggregation on smartnic jbofs. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM '21, page 106–122, New York, NY, USA.

Netronome (2020). Internet.

(PNA), P. P. N. A. (2023). Internet.

Schuh, H. N., Liang, W., Liu, M., Nelson, J., and Krishnamurthy, A. (2021). Xenic: Smartnic-accelerated distributed transactions. In *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*, SOSP '21, page 740–755, New York, NY, USA.

Swamy, T., Rucker, A., Shahbaz, M., Gaur, I., and Olukotun, K. (2022). Taurus: A data plane architecture for per-packet ml. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '22, page 1099–1114, New York, NY, USA.

Viegas, P. B., de Castro, A. G., Lorenzon, A. F., Rossi, F. D., and Luizelli, M. C. (2021). The actual cost of programmable smartnics: Diving into the existing limits. In Barolli, L., Woungang, I., and Enokido, T., editors, *Advanced Information Networking and Applications*, pages 181–194, Cham.

Wei, X., Cheng, R., Yang, Y., Chen, R., and Chen, H. (2023). Characterizing off-path SmartNIC for accelerating distributed systems. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 987–1004, Boston, MA.

Xing, J., Qiu, Y., Hsu, K.-F., Sui, S., Manaa, K., Shabtai, O., Piasetzky, Y., Kadosh, M., Krishnamurthy, A., Ng, T. S. E., and Chen, A. (2023). Unleashing smartnic packet processing performance in p4. In *Proceedings of the ACM SIGCOMM 2023 Conference*, ACM SIGCOMM '23, page 1028–1042, New York, NY, USA.

Xiong, Z. and Zilberman, N. (2019). Do switches dream of machine learning? toward in-network classification. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, pages 25–33.

6. Curriculum Vitae

Marcelo Caggiani Luizelli holds a Ph.D. in Computer Science (UFRGS, 2017) and is an Assitent Professor at the Federal University of Pampa (Unipampa), Brazil since 2017. In 2016, he worked as a visting research at Nokia Bell Labs and Technion University, both in Israel, working with optimization of networking algorithms. In 2022 he was a visiting research at Western Sydney University in Australia. Dr. Luizelli is a CNPq-Brazil Research Fellow and his areas of expertise are network programmability, network optimization, and algorithms. He has authored over 80 papers in leading peer-reviewed journals and conferences related to these topics, and also serves as TPC member for important conferences in these areas.

Francisco Germano Vogt. is a Brazilian PhD student (First year) at the University of Campinas (UNICAMP), Brazil, and works with Information & Networking Technologies Research & Innovation Group (INTRIG). He holds the Bachelor of Computer Science degree from the Federal University of Pampa (UNIPAMPA), Brazil, 2021, and the Master's degree from the University of Campinas, 2023. During his undergraduate project at UNIPAMPA, he worked with In-band Network Telemetry (INT) and positioning of artificial neural networks on the data plane. During his masters project at UNICAMP, he worked using the Inter-Packet Gap (IPG) metric for network monitoring and management in programmable networks. He is involved as a researcher with Ericsson on the project "SMARTNESS 2030: SMART NEtworks and ServiceS for 2030". His ongoing research work proposes an In-network P4-based approach to network monitoring using the Inter Packet Gap (IPG) metric.

Weverton Luis da Costa Cordeiro. has a PhD in Computer Science (UFRGS, 2014). He works as an Associate Professor at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS). His research broadly focuses on Software Defined Networks, Network Function Virtualization, Programmable Data Plans, and Network Security, among others. Dr. Cordeiro is a CNPq-Brazil Research Fellow and is a permanent member of the Postgraduate Program in Computing (PPGC) at UFRGS.

Alberto Egon Schaeffer-Filho holds a Ph.D. in Computer Science (Imperial College London, 2009) and is an Associate Professor and Head of the Graduate Program in Computing at the Federal University of Rio Grande do Sul (UFRGS), Brazil. From 2009 to 2012 he worked as a Research Associate at Lancaster University, UK. Dr. Schaeffer-Filho is a CNPq-Brazil Research Fellow and his areas of expertise are network/service management, network virtualization, network programmability, and security and resilience of networks. He has authored over 100 papers in leading peer-reviewed journals and conferences related to these topics, and also serves as TPC member for important conferences in these areas, including: CNSM, NetSoft, IEEE/IFIP NOMS, IEEE ICC, SBRC and SBSeg. He served as the general co-chair for SBRC 2019, TPC co-chair for IFIP/IEEE IM 2021, TPC co-chair for SBRC 2021 and TPC co-chair for SBSeg 2022. He is also a member of the editorial board of the Journal of Network and Systems Management (Springer) and of the International Journal of Network Management (Wiley). Dr. Schaeffer-Filho served as a Member of the Administrative Board of RNP (2020-2023) and as the Chair of the Special Interest Group on Computer Networks and Distributed Systems (CE-RESD) of the Brazilian Computing Society (SBC) (2019-2021).

Marcos Schwarz is Research & Development Manager. Since 2014, has been working

with RD teams and projects in Ciberinfraestructures, involving advanced internet, network performance monitoring, dynamic circuit networks and orchestration in SDN and Cloud-Native. He holds a B.Sc. degree in Computer Science from Santa Catarina State University (UDESC) in 2011 and a M.S. degree in Computer Engineering from University of São Paulo (USP) in 2014.

Guilherme Mendes Vieira de Matos. currently is a PhD student in the Department of Computer Science at UFSCar, Sorocaba, SP, working at LERIS Research Laboratory. He has a bachelor's degree in Information's Systems from the University Center of the Educational Foundation of Barretos (2016) and a master's degree in Computer Science (2021) from the UFSCar. His main areas of research include Data Centers, Cloud Computing, SDN, Network Programmability, Hardware Acceleration and Network Monitoring.

Christian Esteve Rothenberg is Associate Professor and head of the Information & Networking Technologies Research & Innovation Group (INTRIG) at the School of Electrical and Computer Engineering (FEEC) of the University of Campinas (UNICAMP), where he received his Ph.D. in Electrical and Computer Engineering in 2010. From 2010 to 2013, he worked as Senior Research Scientist in the areas of IP systems and networking, leading SDN research at CPQD R&D Center in Telecommunications, Campinas, Brazil. He holds the Telecommunication Engineering degree from the Technical University of Madrid (ETSIT – UPM), Spain, and the M.Sc. (Dipl. Ing.) degree in Electrical Engineering and Information Technology from the Darmstadt University of Technology (TUD), Germany, 2006. His research activities span multiple layers of distributed systems and network architectures and are often carried in collaboration with academia and industry (e.g., Ericsson, Samsung, CPQD, Padtec, RNP) around the world, leading to multiple open-source networking projects (e.g., RouteFlow, libfluid, ofsoftswitch13, Mininet-WiFi) in the areas of SDN and NFV among other scientific results. Christian has contributed to several international patents, co-authored three books, and over 200 scientific publications, including top-tier scientific journals and networking conferences such as SIGCOMM and INFOCOM, altogether featuring 10,000+ citations (h-index: 35+, i10-index: 80+). His experience as co-author of national and international short courses / tutorials include: SBC SBRC (2010, 2012, 2014), ACM SIGCOMM (2016), IEEE IM (2016), IEEE NetSoft (2017), ENUCOMP (2017), ERIPI (2018), JAI (2023). Recently, Christian has served as Tutorial Co-Chair of ICIN 2022 and IEEE NetSoft 2022. Currently, he is the PI of the FAPESP Engineering Research Center SMARTNESS (SMART NEtworks and ServiceS for 2030) co-funded by Ericsson and expected duration until 2033.

Fábio L. Verdi currently is an Associate Professor in the Department of Computer Science at UFSCar, Sorocaba, SP, leading the LERIS Research Laboratory. He has a bachelor's degree in Computer Science from the Regional University of the Northwest of the State of Rio Grande do Sul (1999), a master's degree in Computer Science (2002), and a Ph.D. in Electrical Engineering with a focus on Computer Engineering (2006), both from the State University of Campinas (Unicamp). He has completed two postdoctoral fellowships, one at Unicamp in 2009 and another one at KTH Royal Institute of Technology in 2022. He has been actively working as TPC and General Chair as well as TPC member of many top conferences such as SBRC, IEEE NetSoft, IEEE NOMS, IEEE CNSM, IEEE SCC, and ACM Sigmetrics SCR. His main areas of research include Data Centers, Cloud Computing, Routing, SDN, Network Programmability, Hardware Acceleration and

Network Monitoring.