X-MAPE: Extending 6G-connected Self-adaptive Systems with Reflexive Actions

M. Liess*, J. Demicoli*, T. Tiedje†, M. Lohrmann†, M. Nickel†, M. Luniak†, D. Prousalis†,
T. Wild*, R. Tetzlaff†, D. Göhringer†, C. Mayr†, K. Bock†, S. Steinhorst*, A. Herkersdorf*
*Technical University of Munich, Munich, Germany
†Dresden University of Technology, Dresden, Germany

Abstract—**Future connected applications require distributed processing on different layers in the network to orchestrate and optimize their behavior. This necessitates a change in the processing architecture, especially regarding the ability to self-adapt to changing circumstances, such as environmental changes or fault scenarios. Self-adaptivity in computing systems was previously often achieved with the MAPE-K concept. However, in real-time and low-latency environments with a distributed, hierarchical processing architecture, the reaction time of the MAPE-K cycle can be limited by its complexity. In this paper, we propose a possible solution to a quicker reaction by introducing X-MAPE: extending self-adaptivity with predefined, configurable, reflexive actions. In essence, X-MAPE aims to construct a low-latency path between monitoring the system and executing actions to influence it, to reflexively react to a changing system state and improving the system's behavior until a more optimal decision can be made in a less time-critical manner. We present possible realizations of this principle on different layers and components of future 6G networks, providing a vision and framework for adaptive computing systems with reflexive reactions.**

Index Terms—**Autonomic Computing, Self-adaptive, 6G**

I. Introduction

The next generation of mobile communications, 6G, is designed to enable a wide range of applications, such as extended reality (XR) or autonomous driving [1]. XR demands both extremely high throughput and ultra-low latency: the former to maintain immersion in a complex environment and the latter to prevent user motion sickness. The latency requirements for autonomous driving are even more stringent, as the system must facilitate split-second decisions to ensure passenger safety. Additionally, the constant variability of the network environment necessitates extreme flexibility. For instance, the participants of autonomous driving, such as vehicles, traffic infrastructure, and pedestrians with smart devices, constantly change cells and network access points as they move. This constant flux can lead to rapidly changing network conditions in terms of bandwidth, latency, and connectivity. To cope with this challenging environment, a concept known as autonomic computing must be introduced in such a network. Key elements of autonomic computing include self-configuration, self-optimization, self-healing, and self-protection.

One way to achieve autonomic computing is to introduce the so called MAPE-K cycle into the system [2]. The acronym stands for *Monitor*, *Analyze*, *Plan*, *Execute*, and *Knowledge*, indicating the four stages of the autonomic control loop and the shared knowledge base that supports them.

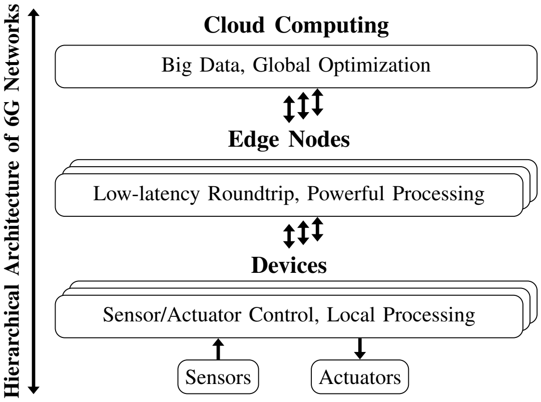
Hierarchical Architecture of 6G Networks
Cloud Computing
Big Data, Global Optimization
Edge Nodes
Low-latency Roundtrip, Powerful Processing
Devices
Sensor/Actuator Control, Local Processing
Sensors
Actuators

Fig. 1. Hierarchical Processing Architecture of 6G Applications

Fig. 1 shows a possible hierarchy of a 6G processing architecture with a device node for local processing of the attached sensors and actuators, an Edge node for low-latency, powerful processing and short-term feedback from the network, and the Cloud infrastructure for Big Data analysis and long-term orchestration. Additionally, a high-level of adaptivity in the employed algorithms and processing nodes of all hierarchical levels is required to react to changing user-, application- and network-behavior, as well as fault scenarios.

This adaptivity is subject to several challenges in 6G networks, which we will express using the terminology of the MAPE-K cycle. Recent trends in autonomic computing have seen a rise in machine learning and generally more complex algorithms to cope with larger and more complex systems. Furthermore, these algorithms are often implemented in software for increased flexibility. Therefore, the *Analyze* and *Plan* stages can become quite computationally intensive and increase processing time. Since device nodes are often constrained in nature, the hierarchical architecture of the network can be used to perform more complex *Analyze* and *Plan* tasks in the Edge. However, the added propagation delay and network latency can again lead to a slower adaption of the MAPE-K cycle. This can have a negative effect on the performance of the adaptive system, since the reaction time increases, thus reducing the chances of a well-optimized system or a successful recovery from a fault.

An abstraction of this problem is that, in certain situations, a complex adaption algorithm can optimize a system's performance in the long-term, but is too slow to enable an immediate

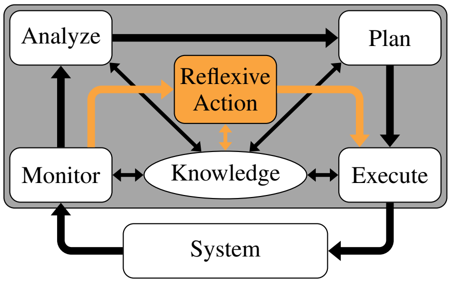

Fig. 2. X-MAPE: MAPE-K cycle extended with reflexive path (highlighted in orange) between *Monitor* and *Execute*.

reaction for a preliminary improvement in performance in case of a fault or other change in system state. A practical example could be a robot encountering an unknown obstacle, requiring an immediate reaction for which extensive processing in the cloud is too slow. Given this abstraction, we can seek inspiration regarding a solution in biology. The human brain has a complex structure, evaluating and learning from countless inputs from sensory neurons and previous responses. This process can be comparatively slow, with a typical reaction time of far more than 100 ms [3]. When this is too slow, reflexes are used to bypass the cognitive thought process and elicit a response almost immediately. They are based on a smaller number of input signals and constrained in their complexity. They are, therefore, fast, specific actions of limited complexity with reduced improvement potential but almost immediate effect. Applying this concept to autonomic computing, the idea is to create a low-latency path between *Monitor* and *Execute* to reflexively react to certain input conditions. Fig. 2 shows the reflex extensions added to the MAPE-K cycle for autonomic computing, which will further be called X-MAPE.

Another work inspired by human reflexes proposed an abstract model of control loops of varying complexity for real-time autonomic systems [4], however, without addressing implications for concrete systems. The conditioned reflex system in [5] focused on reducing the processing overhead in wireless sensor networks by sharing context information between entities, without considering reflexive reactions to system changes or faults. In contrast, X-MAPE targets specific, predefined actions to be executed immediately within hierarchically composed systems when monitoring a certain input condition. Relevant related work will be included in the subsections on the different realizations themselves.

In the remainder of this paper we will in increased detail present the concept and workflow of X-MAPE, in Section II. In Section III we will then present some visions on the deployment and realization of this concept on different layers of the 6G ecosystem, ranging from Memristor Cellular neural networks in analog sensor processing to server resource management in network interface cards (NICs). We will then conclude with a summary and outlook.

II. Conceptual Architecture

Self-adaptivity on different layers of the hierarchical processing architecture has varying requirements in terms of reaction time, knowledge required and regarding the interaction with sensing and acting entities. The wide variety of applications requiring network connection cover a broad range of dynamic behaviors necessitating adaptivity on different levels, much like the human body in biology. Here, reflexes can generally be divided into monosynaptic and polysynaptic. Monosynaptic reflexes, like the knee jerk, only involve one sensor neuron and one motor neuron that are connected with one synapse, usually arced via the spinal cord. A polysynaptic reflex involves at least one interneuron and can contain multiple synapses [6], possibly becoming quite complex and even involving parts of the brain [7]. Interestingly, many reflexes can be inhibited by conscious thought [8] and modulated or learned by classical conditioning [9]. This enables the integration of memory of previous stimuli-response scenarios, which is key to the adaption and creation of reflexes in changing circumstances. While this process can require generations in biology, the programmable nature of computing systems allows reconfiguration to happen in an instant when required.

Therefore reflexes in biology have two important characteristics that shall be mapped to adaptive computing systems. Firstly, reflexes of varying complexity can be used to target the different dynamics of hierarchical systems. While micro-electro-mechanical system (MEMS)-switch based direct connections between sensors and actuators can provide a monosynaptic-like reflex, a polysynaptic reflex could be used to react to a fault requiring information from multiple system parts. Further, while reflexes generally provide a predefined action to be executed, they are by no means fixed. The knowledge that is acquired and shared by a system over time can be used to activate, deactivate or reconfigure reflexes.

While such reflexive extensions are not fixed to be implemented in hardware, the following visions on different realizations in 6G networking systems will focus on enabling low-latency decision-making by configurable hardware reflexes in adaptive systems. The goal is to provide a hardware platform for using reflexes in suitable locations, while the exact function and trigger of these reflexes can be configured.

III. Deployment on different layers of the 6G Ecosystem

The following subsections will cover envisioned realizations of the X-MAPE concept in several components and layers of the 6G processing infrastructure. While the individual modules differ in their approaches and target different components, they all benefit the processing architecture from a systems perspective. An overview of the envisioned deployment is given in Fig. 3 following the hierarchical processing architecture presented in Fig. 1, hereby focusing on the *Devices* and *Edge Nodes*, where low-latency is paramount. Hereby, the left side presents possible extensions in the devices themselves, referring to Subsections III-E and III-F. Moving to the right, the network is traversed, giving the possibility of distributed control (III-B) and more powerful processing either directly in the network (III-C), in general-purpose servers (III-A) or in specialized processing hardware (III-D).

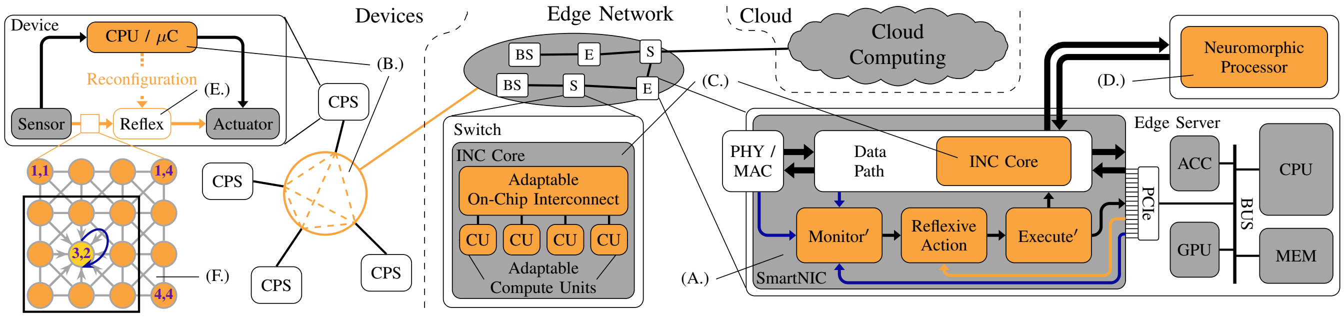

Fig. 3. Overview of envisioned realizations of X-MAPE (highlighted in orange) in different components and layers of an exemplary 6G processing architecture.

A. Network Interface Cards

Many applications connected by 6G have strict latency requirements, thus requiring application-level processing near the user, i.e., in the edge of the network. Due to the frequent changes in human and application behavior, including mobility, environmental changes, etc., the traffic characteristics and processing requirements of an Edge server can vary drastically and rapidly. This requires a NIC that can quickly and energy-efficiently adapt to changing conditions [10].

Concretely, a prerequisite to a performant and energy-efficient operation of a 6G Edge server is the monitoring of the servers state, both regarding the processing resources and traffic characteristics. To fulfill the requirements of all applications, a quality-of-service (QoS) aware processing has to be ensured, e.g. using priority scheduling. To achieve an optimal resource usage, load balancing mechanisms to distribute the processing on the CPU cores [11] can be accompanied by load- and traffic-aware power management to minimize power consumption without jeopardizing the fulfillment of the applications requirements. The principles of X-MAPE can be applied here to tackle these challenges, which is depicted in Part A. of Fig. 3. The Edge server is connected to the network via a SmartNIC. A typical MAPE-K cycle may be executed in software to keep a holistic view and run complex algorithms, but is rather slow in reacting to changes due to the peripheral, memory and processing latencies. Applying the principles of X-MAPE, the SmartNIC can be extended with additional monitoring and execution logic. It can monitor the servers processing resources, inspect and analyze incoming traffic right at the source, and hence make early processing decisions, before all latency-intensive data movement in the server. A minimal reflex latency is ensured by implementing the additional units in hardware on the SmartNIC.

Regarding a concrete implementation, the monitoring logic can include header parsers, packet classifiers and an interface to the PHY to characterize the incoming traffic and driver extensions to monitor the servers processing resources. The reflex can be realized with configurable match-action tables, can be triggered when a certain condition is met or a predefined threshold is surpassed, and can perform actions on the data path, e.g. dropping or forwarding packets to other nodes, as well as influence the processing behavior of the server by making scheduling decisions, steering packets into correct memory regions and performing power management.

B. Adaptive Distributed Cyber-Physical-Systems

In an Industry 4.0 environment, multiple cyber-physical systems (CPS) interconnect to execute complex manufacturing processes. These systems need a robust, flexible, and adaptive communication infrastructure for efficient coordination.

Disruptions such as maintenance or malfunctions can occur, traditionally necessitating lengthy human-led repairs. However, our flexible 6G network architecture can autonomously adapt, enhancing resilience and throughput. Despite this, the MAPE-K loop's computational latency can challenge real-time responses. We address this by incorporating a reflex mechanism. It triggers instantly to maintain uninterrupted CPS operation, while the MAPE-K loop later optimizes system performance. Together, they ensure robust operation by combining immediate action with eventual optimization.

The safety-critical applicability of CPS extends to transportation networks, notably the hyperloop system. With high-speed capsules traveling through near-vacuum tubes, real-time responsiveness is vital. In [12], the hyperloop system integrates the reflex mechanism with an advanced anomaly detection system to quickly respond to potential hazards in the magnetic suspension system, showcasing the applicability of X-MAPE.

Like in the Industry 4.0 environment, the traditional MAPE-K loop, while capable of handling complex system state changes, could be inadequate for dealing with real-time emergencies due to the computational latency of complex *Analyze* and *Plan* stages on constrained devices. Hence, integrating a reflex mechanism with the anomaly detection system provides an efficient solution. This allows the hyperloop system to take immediate action upon the detection of an anomaly, prior to the completion of the full MAPE-K loop.

Therefore, the incorporation of a reflex mechanism in the 6G network architecture, enhancing real-time responses, becomes an invaluable asset across a broad spectrum of CPS, ranging from industrial processes to high-speed transportation systems like the hyperloop. The development of such systems signifies a significant stride towards ensuring continuous operation, safety, and resilience of critical systems, even in the face of unforeseen disruptions.

C. In-Network Computing

The idea of having programmable networks and (parts of) computing algorithms in the network was already proposed in 1997 in the form of active networks [13]. However, due to the lack of computational capacity and programming models for such network devices, it was not successful.

With the advent of Software-Defined Networking (SDN), SmartNICs, programmable switches, and programming languages like P4 [14] using a High-Level Intermediate Representation (HLIR) [15] gave new momentum to this idea.

Reflexes would have to interact with each other or with a central controller, for instance to learn and update rules. However, with an increasing number of connected devices the network traffic would increase drastically. To address this problem, In-Network Computing (INC) can help to reduce the network load by processing the data already in the network instead of sending the complete raw data through the network. Architectural designs for programmable network devices can vary greatly. Although with P4 a vendor- and architecture-independent language for programmable network devices exists, it is still a challenge to leverage features and achieve high performance for these devices.

Therefore, an exploration of programming models that allow these heterogeneous devices to be programmed while leveraging their features without requiring the programmer to manually optimize their code for all of them, is of great importance to make INC feasible.

Furthermore, to provide a flexible *Reflexive Action* path with low latency between *Monitor* and *Execute* which is influenced by the *Knowledge*, new System-on-a-Chip (SoC) designs could be explored that include, for instance, programmable ASICs and/or embedded FPGA (eFPGA).

D. Neuromorphic Computing Hardware

Neuromorphic Computing emerges out of the motivation to understand biological data processing to exploit such mechanisms in technical implementations. In the context of 6G, it has not only been proposed for applications but also to improve communications itself due to its power-efficiency and low-latency [16]. The neuromorphic term is very general and, therefore, used for various systems. In this work we refer to neuromorphic processors as devices that use highly parallel processing on multiple processors and low-latency connection between computing nodes, such as Loihi2, BrainScales or SpiNNaker2 [17] [18] [19].

Reflexes are biological key mechanisms to drastically reduce latency between external influence and a reaction. We define latency as the worst case reaction time between a sensory input until a reaction output is generated. A reflex on neuromorphic multichip-processors can be implemented simpler compared to conventional processing systems. One or more processors on such devices can be fully allocated to process the full X-MAPE cycle. This includes reserved processors for detecting inputs that trigger a reflexive action and, therefore, ensure a low-latency reaction time. Multicasting enables writing of incoming sensory data to multiple memory regions within the system. This ensures that data is close to the relevant processors, following a near-memory-computing approach to reduce access times. Additionally, cores can communicate directly with each other, enabling an efficient and fast implementation of the connections between MAPE nodes and especially the central *Knowledge* node, as depicted in Fig. 2.

E. Reconfigurable Reflex on Device Level

Touching a hot surface triggers a human reflex. Thereby, specialized skin thermoreceptors immediately send electrical signals to the spinal cord, initiating a reflex. These signals travel to interneurons and then to motor neurons, which control the muscles. By bypassing cognitive processes, this reflex ensures a rapid response, reducing the risk of injury. The X-MAPE reflex mechanism can be integrated into technical systems such as robots. When a robotic system encounters a hot surface, it typically lacks the instinct to automatically withdraw its robotic finger. However, by implementing a local reflex mechanism, the robot can protect itself. This low level of abstraction allows a quick and efficient response since the processing and decision-making (i.e., *Analyze* and *Plan* in Fig. 2) take place directly on the device, without delays caused by communication with edge or cloud. E.g., a direct reflex between a sensor (*Monitor*) and an actuator (*Execute*) refers to a mechanism where the sensor detects a certain condition and, based on that information, immediately triggers the actuator to perform a specific action or response (see Fig. 3 Part E.). The hardware realization of the reflex can be done with MEMS-switches triggered at a certain threshold, or microcontrollers triggered by a lookup table or smart sensors with included filter preprocessing capabilities such as Convolutional Neural Networks (CNNs). In addition, the reflex can be reconfigured by reflective analysis and planning of computational layers (*Knowledge*). The reconfiguration can be realized by adjustment of the threshold, a lookup table, or weights of the CNN. This achieves a higher level of adaptability and optimization.

For a simple, miniaturized, and cost-effective reflex hardware solution without compromising functionality, heterogeneous integration with low-cost connection technologies is promising. This involves implementing different functions on highly integrated chips or partitioning a monolithic chip into smaller functional units called ”chiplets”. In the context of 6G applications like smart gloves, modified back-end-of-line (BEOL) technologies are needed for electronic packaging of chiplets. These modifications include sustainable additive methods such as 3D printing, self-assembly processes, and polymer multi-layer technologies. Integration and connection of chiplets can be achieved efficiently and cost-effectively in 2.5/3D stacks, enabling even greater levels of miniaturization.

F. Memristive Cellular Neural Networks

6G networks require power-efficient, high-speed, and low-latency sensory data processing structures. An effective solution for developing such structures is based on Cellular Nonlinear/Neural Networks (CellNNs) as the computational cores integrated with the sensors/actuators that provide the sensory input data, as is exemplary depicted in Part F. of Fig. 3. CellNNs are computational arrays inspired by biology, which consist of locally-coupled elementary dynamical systems called cells. Despite their compact representation, CellNNs demonstrate the ability to establish innovative approaches to information processing through complex dynamics of relatively simple locally coupled dynamical systems.

Non-volatile memristors have the potential to be useful in future electronic applications such as novel signal processing paradigms, combining sensing, computation, and data storage capabilities [20], [21]. To meet the demands of 6G networks, it becomes desirable to deploy fast and energy-efficient processors along with the features offered by memristors, known as Memristive Cellular Neural Networks (M-CellNNs). The integration of memristors into CellNNs offers benefits such as reducing the need for separate memory modules, increasing computational efficiency, and enabling flexible functionality in computing networks. Furthermore, memristors can be utilized as part of cell interconnections, providing flexible functionality in computing networks. These memristor-based connections allow for a wide range of tasks, including synaptic weight adjustments and dynamic adaptation [22]. The reflex process occurs at the device level, highlighting the need of immediate data processing. M-CellNNs can be incorporated directly within a system's sensors, improving overall system performance by enabling localized processing [23]. This method enables real-time, low-latency processing of sensory data, allowing for faster responses and decreasing the need to transfer raw data to a central processing unit.

By integrating M-CellNNs into the X-MAPE mechanism, the *Monitor* phase benefits from their ability to assess data and trigger system responses directly in the *Execute* module when necessary. Therefore, as indicated in the Fig. 3, their integration with the sensors can play a role in expanding the mechanism to *Reflexive Action*. Training of weights within M-CellNNs can be achieved by connecting the local device to the cloud. More sensors in a 6G network offer increased data collection, and the cloud-based processing leads to improved weight sets. The updated weights are quickly implemented using the properties of memristors, improving M-CellNN and X-MAPE mechanism performance.

IV. Conclusion and Outlook

Many applications require the ability to quickly adapt to changing circumstances on several layers in modern connected systems, especially the 6G ecosystem. X-MAPE extends the MAPE-K autonomic computing framework with a low-latency, reflexive path between monitoring the system and executing actions that benefit the system as soon as possible. Several visions on realizations of this concept on different layers of 6G networks were described, spanning many components directly from the sensor to deep into the network. Further, the integration and interplay of X-MAPE between different layers and components was sketched. In future work, concrete implementations for each presented component shall be explored and evaluated in detail. An initial individual evaluation should be followed by a combination and integration of all components, exploring the benefit of applying X-MAPE to a hierarchical, connected processing infrastructure.

Acknowledgment

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK001K and 16KISK002.

References

[1] T. Taleb, R. L. Aguiar *et al.*, "White paper on 6g networking," 2020.

[2] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[3] A. Jain, R. Bansal *et al.*, "A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students," *International Journal of Applied and Basic Medical Research*, vol. 5, no. 2, p. 124, 2015.

[4] R. Sterritt and M. Hinchey, "Adaptive reflex autonomicity for real-time systems," *Innovations in Systems and Software Engineering*, vol. 5, no. 2, pp. 107–115, 2009.

[5] M. Hussain, M. F. Shafeeq *et al.*, "Cram: a conditioned reflex action inspired adaptive model for context addition in wireless sensor networks," *Journal of Sensors*, vol. 2016, 2016.

[6] B. Hopkins, E. Geangu, and S. Linkenauger, *The Cambridge Encyclopedia of Child Development*, 2nd ed. United States: Cambridge University Press, 2017.

[7] R. Schondorf, "Chapter 84 - assessment of sudomotor function," in *Primer on the Autonomic Nervous System (Third Edition)*, D. Robertson, I. Biaggioni *et al.*, Eds. San Diego: Academic Press, 2012, pp. 409–411.

[8] F. Verbruggen, M. Best *et al.*, "The inhibitory control reflex," *Neuropsychologia*, vol. 65, pp. 263–278, 2014.

[9] P. I. Pavlov, "Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex," *Annals of neurosciences*, vol. 17, no. 3, p. 136, 2010.

[10] AMD, "White paper: Adaptive smartnics for future data center architectures," https://www.xilinx.com/content/dam/xilinx/publications/white-papers/adaptive_smartnic_whitepaper.pdf, Jul. 2023.

[11] F. Biersack, K. Holzinger *et al.*, "Priority-aware inter-server receive side scaling," in *Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. IEEE, 2023, pp. 51–58.

[12] J. Demicoli, L. Prenzel, and S. Steinhorst, "Autonomous hyperloop control architecture design using mape-k," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2023, pp. 1–6.

[13] D. Tennenhouse, J. Smith *et al.*, "A survey of active network research," *IEEE Communications Magazine*, vol. 35, no. 1, pp. 80–86, Jan. 1997.

[14] "P4˜16˜ Language Specification," https://p4.org/wp-content/uploads/2022/07/P4-16-spec.html, Jul. 2022.

[15] H. Stubbe, "P4 compiler & interpreter: A survey," *Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM)*, vol. 47, 2017.

[16] J. Hoydis, F. A. Aoudia *et al.*, "Toward a 6g ai-native air interface," *IEEE Communications Magazine*, vol. 59, no. 5, pp. 76–81, 2021.

[17] G. Orchard, E. P. Frady *et al.*, "Efficient neuromorphic signal processing with loihi 2," *CoRR*, vol. abs/2111.03746, 2021.

[18] C. Pehle, S. Billaudelle *et al.*, "The brainscales-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, 2022.

[19] C. Mayr, S. Höppner, and S. B. Furber, "Spinnaker 2: A 10 million core processor system for brain simulation and machine learning," *CoRR*, vol. abs/1911.02385, 2019.

[20] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[21] D. B. Strukov, G. S. Snider *et al.*, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.

[22] Z. Yang, L. Zhang *et al.*, "Hardware-mappable cellular neural networks for distributed wavefront detection in next-generation cardiac implants," *Advanced Intelligent Systems*, vol. 4, no. 8, p. 2200032, 2022.

[23] R. Tetzlaff, A. Ascoli *et al.*, "Theoretical foundations of memristor cellular nonlinear networks: Memcomputing with bistable-like memristors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 2, pp. 502–515, 2020.