



## Article

# A Survey of MPSoC Management toward Self-Awareness

Guillermo Gonzalez-Martinez <sup>1</sup>, Remberto Sandoval-Arechiga <sup>1,2,\*</sup>, Luis Octavio Solis-Sanchez <sup>1</sup>,  
Laura Garcia-Luciano <sup>1</sup>, Salvador Ibarra-Delgado <sup>1,2</sup>, Juan Ramon Solis-Escobedo <sup>1</sup>,  
Jose Ricardo Gomez-Rodriguez <sup>1,2</sup> and Viktor Ivan Rodriguez-Abdala <sup>1,2</sup>

- <sup>1</sup> Posgrado en Ingeniería y Tecnología Aplicada (PITec), Universidad Autónoma de Zacatecas, Av. Ramón López Velarde, 801, Col. Centro, Zacatecas 98000, Mexico; ing.memogm@uaz.edu.mx (G.G.-M.); lsolis@uaz.edu.mx (L.O.S.-S.); lgl.hardware@uaz.edu.mx (L.G.-L.); sibarra@uaz.edu.mx (S.I.-D.); juan.solis@uaz.edu.mx (J.R.S.-E.); jrgrodri@uaz.edu.mx (J.R.G.-R.); abdala@uaz.edu.mx (V.I.R.-A.)
- <sup>2</sup> Centro de Investigación, Innovación y Desarrollo en Telecomunicaciones (CIDTE), Universidad Autónoma de Zacatecas, Av. Ramón López Velarde, 801, Col. Centro, Zacatecas 98000, Mexico
- \* Correspondence: rsandoval@uaz.edu.mx; Tel.: +52-492-9256690 (ext. 4009)

**Abstract:** Managing Multi-Processor Systems-on-Chip (MPSoCs) is becoming increasingly complex as demands for advanced capabilities rise. This complexity is due to the involvement of more processing elements and resources, leading to a higher degree of heterogeneity throughout the system. Over time, management schemes have evolved from simple to autonomous systems with continuous control and monitoring of various parameters such as power distribution, thermal events, fault tolerance, and system security. Autonomous management integrates self-awareness into the system, making it aware of its environment, behavior, and objectives. Self-Aware Cyber-Physical Systems-on-Chip (SA-CPSoCs) have emerged as a concept to achieve highly autonomous management. Communication infrastructure is also vital to SoCs, and Software-Defined Networks-on-Chip (SDNoCs) can serve as a base structure for self-aware systems-on-chip. This paper presents a survey of the evolution of MPSoC management over the last two decades, categorizing research works according to their objectives and improvements. It also discusses the characteristics and properties of SA-CPSoCs and explains why SDNoCs are crucial for these systems.

**Keywords:** multi-processor system-on-chip; MPSoC management; self-awareness; self-aware cyber-physical systems-on-chip; software-defined networks-on-chip; survey



**Citation:** Gonzalez-Martinez, G.; Sandoval-Arechiga, R.; Solis-Sanchez, L.O.; Garcia-Luciano, L.; Ibarra-Delgado, S.; Solis-Escobedo, J.R.; Gomez-Rodriguez, J.R.; Rodriguez-Abdala, V.I. A Survey of MPSoC Management toward Self-Awareness. *Micromachines* **2024**, *15*, 577. <https://doi.org/10.3390/mi15050577>

Academic Editor: Arman Roohi

Received: 12 March 2024

Revised: 16 April 2024

Accepted: 23 April 2024

Published: 26 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As technology advances, the need for adequate system management becomes increasingly important. This is especially true in the case of Multi-Processor Systems-on-Chip (MPSoCs), where the number of processing elements must increase to keep up with market demands. The Internet of Things [1,2], artificial intelligence, and cloud-based digital systems [3,4] are just a few examples of technologies that have driven this need. However, each application has specific requirements, making MPSoC management a complex challenge. Adding more processing and communication resources leads to energy consumption, temperature variations, and vulnerability to different failures. Some platforms, such as the Kalray MPPA-256 [5], the Adapteva Epiphany [6], and the Sunway [7], address these issues through distributed, scalable, and heterogeneous systems. While these chips have been successful in the industry, they lack the organizational management structure necessary for migration to more powerful systems, such as a self-aware system-on-chip.

Managing an MPSoC can be challenging due to the ever-increasing demand for enhanced capabilities and the dynamic nature of new and upcoming applications. Expanding the capabilities of an MPSoC involves increasing the resources, components, and metrics it has to control, either in number or complexity. Therefore, an efficient management approach is necessary to handle functionality aspects at various levels. This includes physical

elements like processing elements, memories, ports, communication, monitoring infrastructures, and nonphysical factors like process tasks, utilization time, data, and bandwidth.

This situation highlights the need for an efficient administration that can manage various aspects in a coordinated manner to achieve the system's objectives. As a result, the management of an MPSoC should contemplate the implementation of optimization engines, i.e., control and supervision techniques and/or protocols aimed to ensure an efficient performance.

Managing the communication infrastructure of an MPSoC is critical to its overall performance. One area of significant research is the interconnection of multiple processing elements through Network-on-Chip (NoC). The NoC infrastructure involves routers that connect processing components, providing excellent scalability to MPSoCs [8]. However, with the diversity of applications and heterogeneity of new systems, the communication infrastructure must efficiently handle dynamic patterns and workloads. Poor management of NoC can lead to significant problems, such as congestion, thermal hot spots, deficient performance, and missing deadlines, so network management is essential. It should control resources such as routers, interfaces, buffers, links, packets, transmission rates, and waiting times. The system's active supervision requires efficient implementation and control of optimization engines at various network layers.

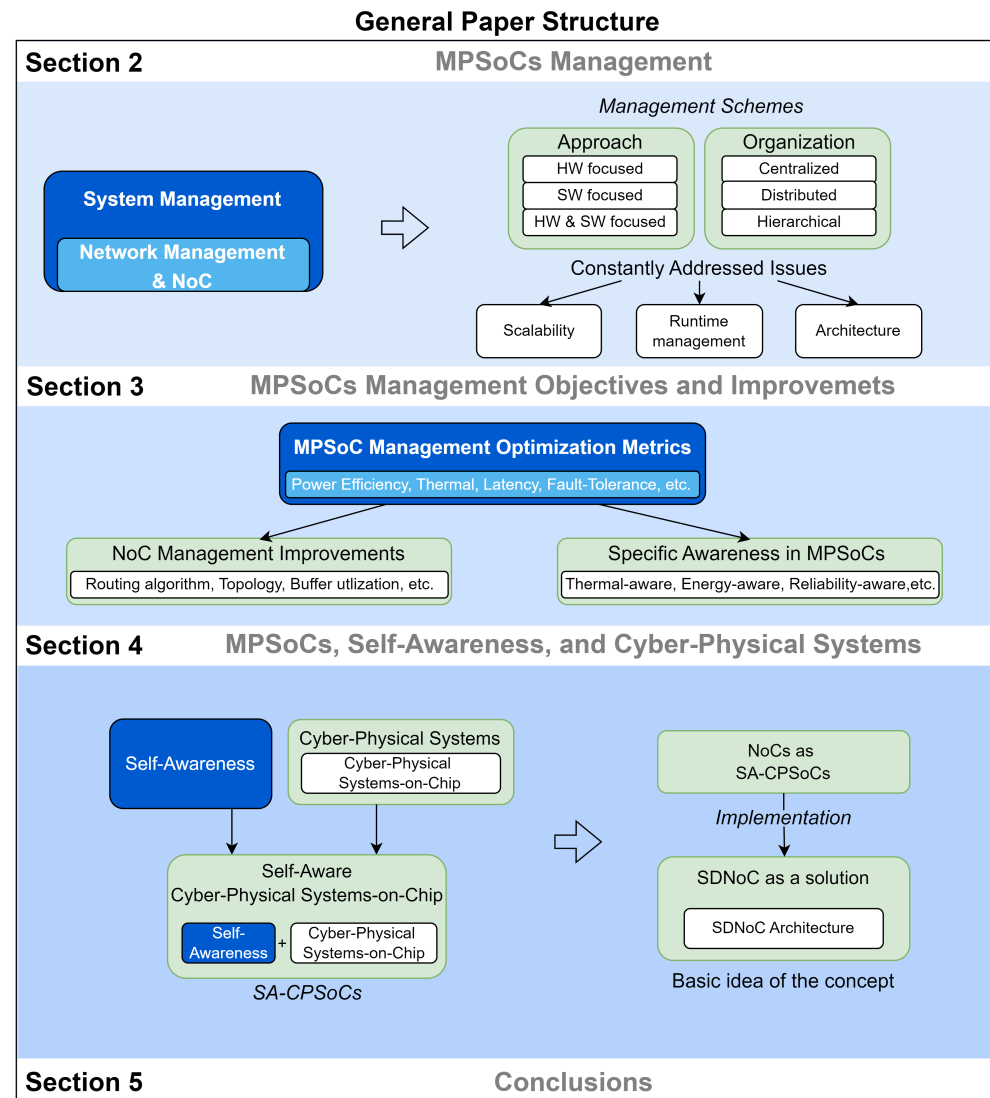
When analyzing global system administration, it is also important to consider network administration due to their mutual interconnectedness; a network process is not independent of an application process. However, the architecture and abstraction capacity of MPSoCs allow for separate analyses of different management types while still incorporating dynamic adaptability, intelligence, and proactivity. Ignoring communication infrastructure in global management can lead to poor performance and high energy consumption [9]. Intelligent management involves monitoring and configuring control functionalities through various services [2,9]. As such, researchers have studied techniques and tools to achieve flexibility, reconfigurability, and adaptability at runtime.

Several management schemes proposed involve novel concepts like cognitive networks, self-aware systems, and Software-Defined Network-on-Chip (SDNoC) systems [2,10–14]. Each scheme has different structures, approaches, scopes, and optimization objectives. However, there is a research gap in this context, as no generalized modular framework is available. To address this gap, a software-based management framework is required to offer services for reuse and facilitate the development of robust embedded systems.

In this paper, we surveyed the literature on the management of MPSoC and its potential for future development. Over the past two decades, we have compiled research that specifically focuses on the management of MPSoCs. The proposed schemes have been classified based on their architecture, approach, objectives, and improvements. Our taxonomy highlights the most researched management areas and identifies those that require more attention to help overcome challenges posed by new technologies. Furthermore, we discuss the concepts of self-awareness and cyber-physical systems and their relevance to MPSoCs. Lastly, we emphasize the importance of network management and its impact on overall system management. We also suggest that the concept of SDNoC could potentially be advantageous in meeting the demanding requirements of new and future MPSoCs.

The rest of the paper is organized as follows: In Section 2, we provide an overview of MPSoC management, including its concept, characteristics, and various management approaches and organizations described in the literature. We also propose a classification based on important issues that have influenced the development of MPSoCs. Section 3 classifies and analyzes different research works on proposed management schemes and their optimization objectives. We classify NoC-related works according to their main optimization metric and the most common improvement areas of NoC management. We also classify works with specific awareness, or that implemented self-x properties (focusing on adding different characteristics to the system to manage and perform processes without third-party intervention). Section 4 discusses the evolution and development of self-awareness and cyber-physical systems and their relationship, integration, and challenges

in MPSoCs. The end of this section explains how a structured SDNoC architecture can help develop Self-Aware Cyber-Physical Systems-on-Chip (SA-CPSoC) through network-based system management. Finally, in Section 5, we conclude our work. Figure 1 shows the general paper structure from Section 2.



**Figure 1.** General paper structure.

## 2. MPSoC Management

When designing an MPSoC, it is essential to manage all the interconnected processing elements within the system. With hundreds or thousands of elements, network management becomes a critical issue. While some platforms on the market offer solutions, they still require an organizational management structure capable of hosting features to monitor and control parameters within a system, aware of its state, environmental interactions, behavior, and goals. These parameters include, for example, power distribution, thermal events, fault events, security attacks, link bandwidth, routing, or traffic distribution. The following subsections present different management approaches, organizations, and issues addressed in the MPSoC research.

### 2.1. System Management

Efficient system management of an MPSoC is crucial for its overall functionality and performance. It involves optimizing processes required by applications, utilizing both

hardware and software resources available in the system. These resources are complex and varied, with different levels of abstraction, including processing elements, specific process tasks, communication infrastructure, and others. Properly managing these resources involves controlling various actions like task mapping, scheduling, migration, element mapping, data distribution, and memory access [15,16].

To improve MPSoC performance, system management implements techniques and optimization engines, ranging from simple actions like turning an element off and on to complex algorithms and properties that enable self-awareness. Researchers in this area focus on specific problems like power and temperature management, QoS management, or network management to improve system management schemes [17,18].

## Network Management

Managing the network within an MPSoC environment is crucial. Network management involves gathering information from the communication infrastructure, analyzing it, and taking corrective or preventive measures. It is a complex task to manage a network of hundreds or thousands of processing elements, and it becomes even more challenging when there is a need for runtime adaptability to handle a modern system's workload variability. The NoC paradigm helps differentiate between computational and communication problems. However, proper network management is essential to prevent the NoC from becoming the bottleneck of system performance [2]. Hence, network management requires new control strategies that enable multiple processing elements to interact appropriately, access system resources, manage processes that require shared resources, and adapt to the environment's variability at runtime.

The network management schemes depend on the type of communication infrastructure used, such as point-to-point links (P2P), interconnection buses, interconnection crossbar switches, or NoCs to interconnect the processing elements of an MPSoC. NoC is one of the most widely accepted MPSoC interconnection architectures. It uses traditional router and packet switch network concepts at the intrachip level. NoC architecture outperforms its counterparts in many aspects, especially regarding flexibility, scalability, and energy efficiency [8].

### 2.2. Management Approaches

Management can implement different strategies using hardware, software, or both, depending on objectives, optimization protocols, and processes.

#### 2.2.1. Hardware-Focused

Hardware-focused management schemes aim to introduce hardware elements with minimal or no use of software. These elements may include specialized monitoring agents or other dynamic management components. Hardware-focused systems are typically faster than software-based systems, as they can perform multiple tasks in parallel [19]. This approach automates management processes such as path switching, where processing speed is more favorable than the overhead that software-based implementations may introduce. However, implementing hardware-focused schemes can also lead to critical problems, such as increased area consumption, incompatibility caused by the addition of control lines, and the need for redesigning that may require longer development times [10]. Thus, the designer's community aims to minimize hardware overhead by focusing its research efforts on developing effective hardware elements with minimal area consumption.

#### 2.2.2. Software-Focused

The implementation of software-based management systems is designed to optimize processes using software routines. This can be achieved by adding pure software agents or making adjustments at the operating system (OS) level. Although this approach adds communication and computation overhead, there are certain management processes where these overheads are unavoidable, such as congestion and flow control, which require software routines and the exchange of control messages [10]. Additionally, using silicon logic

gates is generally cheaper than wires, and the development of software implementations usually involves less effort and time.

### 2.2.3. Hardware and Software Focused

A management scheme can have a specific focus on either hardware or software, each with its own set of advantages and disadvantages. Hardware-focused strategies tend to be faster, but the ever-changing technological systems requirements are also faster than the time it takes to develop hardware. On the other hand, software-focused schemes tend to be slower, but they have the advantage of being faster to implement. Due to the challenges posed by new MPSoCs, many research papers have implemented management schemes that combine both approaches to leverage the benefits of each system. These papers establish management protocols where the congested parts performing automated tasks use a hardware approach, while the software-based approach is used for parts that require constant changes. By building the management scheme protocols offline and re-configuring them at runtime, software allows for systems with dynamic requirements to be optimized [20]. This is especially important given the dynamism, flexibility, and harsh requirements of new MPSoCs, which drive changes in embedded systems. Therefore, studies have combined software- and hardware-focused implementations to achieve different optimization objectives with low overheads in new MPSoCs [21,22].

## 2.3. Management Organization

System management is carried out differently depending on the control assignment of the management entity/entities. How management is organized significantly impacts important characteristics such as scalability and ease of implementation. Centralized, distributed, or hierarchical management schemes are commonly used in this context.

### 2.3.1. Centralized

In centralized management, a central entity is responsible for overseeing the entire managed system. It executes various control functions and optimization engines from a central location. Generally speaking, centralized management offers several advantages, such as deadlock avoidance due to the network overview, greater fairness in resource utilization between elements, greater simplicity of data forwarding entities, reduction in network overhead, and ease of obtaining performance metrics [23]. However, its most crucial disadvantage is the scalability problem, and its use is limited to small MPSoCs [24]. Centralized management can also reduce the system's long-term reliability since the constant demands of attention to different actions, such as mapping or event monitoring, make it susceptible to failures [24,25].

### 2.3.2. Distributed

Distributed management aims to overcome centralized management's bottleneck and scalability problems [14]. To achieve this, the managed system is spatially or logically partitioned, i.e., the MPSoC can be divided into different regions (clusters), each with its management entity, or there can be one management entity per application. This strategy helps improve the system's reliability and QoS by lightening the burden on manager entities. However, distributed managers also bring drawbacks, such as access to input and output devices that remain centralized entities, the control and allocation of cluster sizes, or the number of applications running on an MPSoC [24].

### 2.3.3. Hierarchical

In the management field, the architecture can be centralized or distributed and may include a hierarchical organization. This organization categorizes the elements of the architecture into different operational levels and defines a hierarchy for each level. Elements at each level only communicate with those above or below their class. This hierarchical structure provides autonomy to various entities, thus enhancing their independence charac-



teristics within the system. Hierarchical management schemes are designed to help manage ultra-large-scale MPSoCs [18,26].

#### 2.4. Constantly Addressed Issues

In the design and development of MPSoCs, some aspects of their evolution must be considered. These include the constantly growing scalability issues, the runtime adaptability required by the new systems, and the paradigm changes in architectures that this demands. This approach opens the door to new challenges, such as adding self-adaptation and intelligence to future MPSoCs.

##### 2.4.1. Scalability

A key feature that current and future MPSoCs must offer is high scalability. With the increasing demand for higher performance and other new application requirements, the trend for embedded systems is to add more processing elements. However, providing high scalability can become a significant challenge for MPSoCs when talking about hundreds or thousands of processing elements. Therefore, it is necessary to consider adequate management of the MPSoC resources to ensure scalability [15], together with a layered architecture that isolates different problems to be solved independently [2].

High scalability comes with other requirements, such as power, temperature, and reliability, which become even more significant challenges for designers. In addition, incorporating intelligence in conjunction with online adaptation demands architectural improvements in new and future MPSoCs. Thus, when talking about a system with self-adaptability, scalability significantly impacts operational efficiency and can make the system objectives more straightforward to achieve [27]. Several investigations aim to increase or ensure scalability in MPSoCs. Network management is a highly investigated topic because it can become a system performance bottleneck. The recent paradigm of SDNoCs showed good scalability and network resource management. These characteristics of SDNoCs are due to their flexibility, reliability, and dynamic adaptability [2,4,9].

##### 2.4.2. Runtime Management

Today's systems need to be flexible and adaptable to the constant changes that the dynamic behavior of new applications demands. In addition, they must provide the highest possible efficiency by taking care of the performance metrics that the application requirements dictate. Several investigations have developed schemes that allow on-the-fly dynamic management, whose objective is to provide optimization engines capable of online adapting to dynamic changes in the environment, such as varying workloads. As a result, this type of runtime-managed system has become one of the most important and crowded research areas [20]. One of the challenges for new systems is appropriately managing the available resources to perform proactive optimization, such as monitoring infrastructures, triggering events, decision making, learning algorithms, etc. Systems must perform these actions while making the appropriate trade-offs to meet the various requirements. All these actions involve the supervision of different adjustable parameters that modify the system behavior, so they must be performed at runtime to achieve optimizations according to the environment [28].

When the adaptability in MPSoC began to be studied, most research contemplated that events coming from external entities, such as the application layer or even a human operator, triggered the adaptation actions. However, current and future MPSoCs require the system to identify these events and initiate the adaptation processes, leading to self-adaptation [29].

##### 2.4.3. Architecture

While some research papers rely on traditional architectures in which they implement their proposed management of various resources, others have presented new ideas at the architectural level to improve the overall or point performance of an MPSoC. Within the diversity of research papers, some focus on making modifications at the hardware level only,

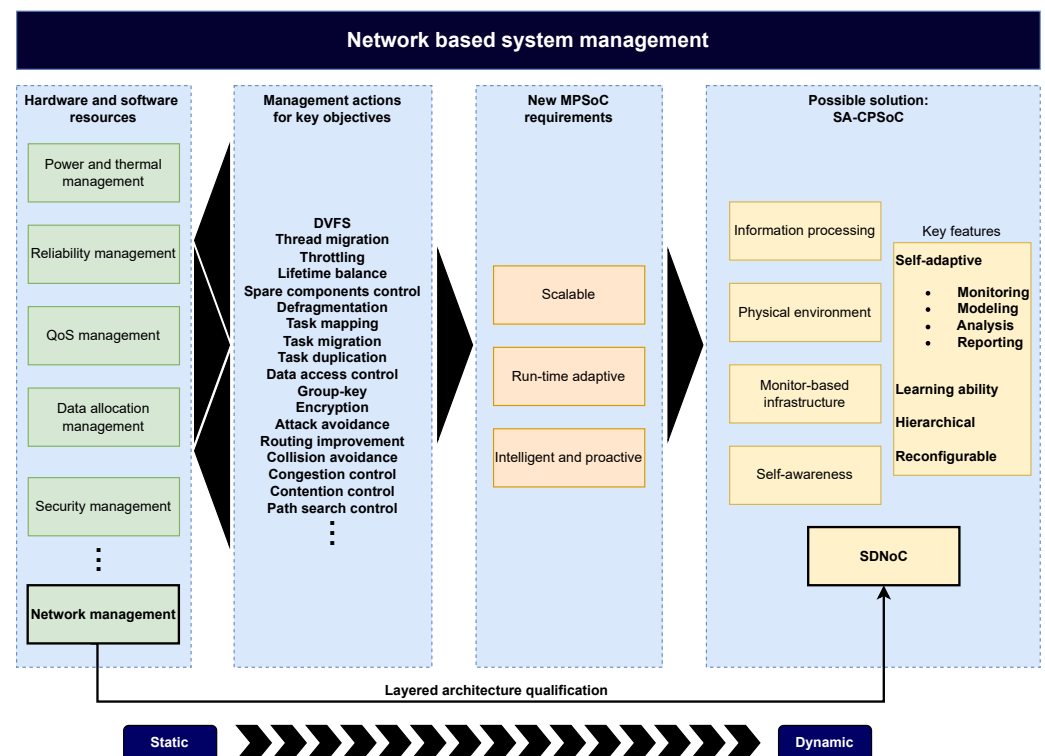
and others at the software level only, but most concentrate on implementing innovations that involve hardware and software. Likewise, the new dynamic requirements and the high scalability of emerging MPSoCs demand architectural improvements at different levels. One of the most significant is related to the NoC intercommunication infrastructure. Thus, the system's architecture must contemplate new management, control, and supervision schemes to meet the new expectations [30].

## 2.5. Evolution of MPSoC Management

MPSoC management aims to create highly dynamic environments where the constant variation of application processes demands versatile handling of hardware resources and task coordination. As the number of processing elements incorporated within an MPSoC increases, there is a need for resource management and supervision with more outstanding capabilities [30] to handle the higher power and temperature density [27], as an example. These new paradigms challenge MPSoC management, requiring different goals regarding management subdivisions such as energy, power, temperature, system reliability, QoS, security, network, etc.

Recent research into Self-Aware Cyber-Physical Systems-on-Chip (SA-CPSoCs) has demonstrated that they can solve the challenges of new and future MPSoC developments. The SA-CPSoC paradigm incorporates critical features such as self-aware, self-adaptive, learning, and reasoning capabilities within an infrastructure that enables excellent monitoring and actuation capabilities over the physical and virtual environment.

Network resources management is a fundamental part of any system, and it can be a determining factor for the optimal management of the entire system. In this context, the hierarchical layered architecture paradigm of Software-Defined Networks-on-Chip (SDNoCs) can be a component that helps in developing and evolving MPSoCs towards SA-CPSoCs through network-based system management. Figure 2 shows this evolution based on the new fundamental requirements of MPSoCs and the critical features of the possible solution represented by SA-CPSoCs.



**Figure 2.** New fundamental requirements for a Self-Aware Cyber-Physical System-on-Chip through network-based system management.

## 2.6. Summary

Modern MPSoCs present a range of new challenges for designers striving to maximize their capabilities. One of the key requirements for these systems is to add runtime adaptability while being self-aware of their state, environment, behavior, and goals. To meet these challenges, it is essential to have a management and control layer dispersed across several abstraction levels that can act according to the system's needs, leveraging the most suitable management characteristics regarding approach, organization, and implementation status. Table 1 provides a classification of some of the most relevant research of the past two decades that focused on management issues related to MPSoCs and addressed the constantly evolving problems in this field. Although achieving greater scalability has always been one of the objectives of MPSoCs, a physical limit has been reached. Thus, it is necessary to use the available resources more efficiently and optimally based on the system's requirements at any given time. Based on Table 1, about 65% of the research focused on adding runtime capabilities to enable the system to handle constraints and dynamism. While most of these research papers still preferred centralized schemes, there has been an increase in developments with hierarchically distributed management schemes since 2010.

These challenges have led to the design and development of new proposals for architectural improvements. Therefore, overall system management that actively involves monitoring and control strategies of the communication infrastructure may be the right path towards highly scalable MPSoCs with self-aware and self-adaptive capabilities.



**Table 1.** Classification of management research papers of the last twenty years.

		Organization			Focus		
		Centralized	Distributed	Hierarchical	Hardware	Software	Hardware & Software
Attendable Issue	Scalability	[9,23,31–41]	[4,9,15,24,31,38,42–55]	[18,27,31,43,45,56,57] [2,15,32,37,38,47,51,53]	[34,39,53,58]	[24,31,42,45,47,59] [15,37,38,40,49,54]	[18,27,43,57,60] [2,33,41,48,51,55,61,62]
	Runtime Management	[25,31,63–71]	[24,31,42,43,69,99–104] [26,30,46,47,84,105–107]	[31,65,69,101,115,116] [26,30,43,73,117,118]	[65,73,103,120] [88,110,121]	[16,31,59,63,64,122–125] [24,26,42,74,75,77,78,116]	[25,43,66,68,69,100–102] [17,72,76,79,104,105,107]
		[11,17,21,35,38,72–98]	[29,38,51,55,89,108–111] [14,15,53,94,112–114]	[28,38,47,51,53,93,108] [11,15,97,113,114,119]	[10,53]	[47,81,84,85,117,126–129] [15,38,91,98,108,109,112–114]	[28,30,55,82,83,86,87,92] [20,21,51,93,97,111,119,130]
	Architectural Improvement	[68,73,131–133] [17,33,80,82,83]	[99–101,131,137,138] [22,43,45,105,107,139]	[18,101,115,132,133,137] [22,28,43,45,73,138]	[58,73,142] [121,139,140]	[16,125,131] [45,143]	[18,19,100,101,132,133,137] [17,22,43,68,105,107,144]
		[1,87,93,134,135]	[52,53,55,113,140,141]	[53,93,97,113,119]	[1,53,141]	[40,98,113]	[28,33,55,82,83,87,134]
		[21,40,97,98,136]					[12,21,61,93,97,119]

### 3. MPSoCs Management Objectives and Improvements

Over the last twenty years, research papers have been primarily focused on developing and implementing techniques and procedures to improve specific optimization metrics. However, the MPSoC management is evolving towards making the system capable of simultaneously fulfilling multiple optimization objectives, paying special attention to network processes. In this context, some researchers have been working on developing “awareness” by adding monitoring and actuation capabilities to solve specific issues within the MPSoC environment. While these works were not a complete conception of “self-awareness” within MPSoCs, they serve as a crucial motivational precedent. These papers showcased systems with specific awareness to assist certain processes and improve optimization metrics. In this section, we present an analysis and classification of the optimization metrics that have been most worked on to improve the MPSoCs management. We discuss the concept of an NoC and its management and its important role in the performance of an MPSoC. We show a classification of the improvements made to the NoC environment to address the different optimization metrics of MPSoCs. Finally, we also present a classification of different specific awareness that have been worked on in MPSoCs.

#### 3.1. MPSoCs Management Optimization Metrics

Several optimization metrics need to be considered to evaluate the performance of an MPSoC. In this paper, we considered the optimization metrics that have become more popular in the last twenty years, according to the literature. The most common metrics we focused on were power efficiency, temperature, fault tolerance, latency, throughput, security, QoS, execution time, and area.

- **Power efficiency:** One of the most relevant and researched aspects in the last decades is the energy consumption of embedded systems. The technological demands of new platforms have led to the integration of multiple processing elements within the same chip since they provide a level of parallelism that allows solving of the performance requirements of increasingly complex applications [145]. An NoC typically interconnects an MPSoC, which consumes a significant portion of the system power, so power consumption has become a crucial performance metric when designing [2]. An increase in specific parameters is required to meet more strict performance requirements, for example, higher operating frequencies. These demanding conditions and the workload variability of new systems increment power consumption and heat dissipation. Therefore, efficient management of these aspects has become vital in modern designs, especially in battery-operated mobile systems.
- **Thermal:** The on-chip temperature control of modern MPSoCs has become crucial because of its short- and long-term implications. These implications are related to high-temperature variations, which could severely affect the system’s reliability and performance [146]. These thermal conditions are especially detrimental to more temperature-sensitive systems such as optical NoCs [147]. Since conventional on-chip cooling is unavailable due to cost and space constraints, researchers are developing techniques to manage the temperature of SoCs. These management schemes also help to increase the tolerance to permanent failures, extending the lifespan of the components since the temperature is one of the leading agents that accelerates the aging effects of the SoCs [86]. Furthermore, these management techniques must be robust enough to deal with the space and time temperature distribution that the complexity of the new system NoC imposes [148].
- **Fault tolerance:** An MPSoC is subject to different failures affecting processing and communication links. System reliability is affected by the faults that the system may incur, so many researchers have designed architectures and management schemes to anticipate and avoid certain types of failures. The types of failures identified within systems-on-chip, especially in new MPSoCs, fall into three main categories: transient, permanent, and intermittent faults [149]. These failures are caused by effects such as soft (cosmic) errors, crosstalk, electromagnetic interference (EMI), intersymbol interfer-

ence, noise, electromigration, and aging of materials [150,151]. Transient faults have a random behavior occurring in one or several execution cycles, while permanent are due to wholly damaged components that cause logic faults or operation delays. Intermittent faults have repetitive behavior and occur in the same place [149]. Several MPSoCs include spare structures to tolerate some of these failures, leveraging the increased number of processing elements. However, the increased number of processing elements sets new challenges, which makes combining management schemes with runtime system monitoring and actuation necessary to add fault tolerance.

- **Latency:** Communication latency within networks is defined as the time it takes for a packet to go through the network from the source node to the destination node, measured in clock cycles [2]. Latency can also denote the time it takes for some process to be performed from start to finish. For example, path-finding latency refers to the time it takes for the system to define communication paths in a circuit-switched scheme [14].
- **Throughput:** In a communication network, throughput is the packet rate delivered by the network, measured in bits per clock cycle. This metric is based on the count of packets reaching their destination within a given time interval for each source–destination link pair. Throughput is also defined as the maximum load the physical network can handle. Current MPSoCs demand higher requirements for applications running task parallelism with intensive information exchanges [86]. Thus, the system must offer throughput guarantees to meet the deadlines incurred by demanding applications [115]. Resource management focused on controlling certain variables, such as congestion or network traffic, can significantly benefit this performance metric.
- **Security:** Security has taken an important role in recent years within the MPSoC environment. New paradigms, such as IoT, seek the massive integration of devices sharing resources, making them more vulnerable to malicious attacks. Most MPSoCs are interconnected by NoCs that have access to all system resources and information, so most attacks are aimed at corrupting the NoCs through malicious software. This malicious software degrades the overall performance of the system and its services, breaches sensitive information, and can even cause failures in its components, such as routers or switches. For this reason, researchers are developing various management schemes to manage particular resources more efficiently. These schemes include the use of private keys and agreements, runtime monitoring of network traffic, and dynamic adaptability of the system to provide support against the most common attacks such as [37] denial of service attacks (DoS attacks), distributed time attacks (DTA), spoofing, tampering, repudiation, information disclosure, or privilege elevation.
- **QoS:** Quality of service encompasses a series of specific requirements linked to optimizing particular metrics for a given expected performance. Therefore, QoS is related to providing certain guarantees for specific requirements such as reliability, bandwidth, or latency in scenarios involving restrictions and limitations [150].
- **Execution time:** Many applications require performing several subprocesses simultaneously within an MPSoC interconnected by an NoC. The execution time and energy efficiency of these subprocesses are vital for real-time applications and various domains. The execution time of these subprocesses depends on the general state of the system, the critical subprocess, the available resources, and their management [40,85]. Resource management can directly influence the execution time of various applications. A way to achieve this is by employing self-awareness and monitoring-based frameworks to add adaptability to the system [66,85]. Another method can be migrating tasks to contiguous processing elements [127], or managing shared data in memory (Scratchpad-memory) [40].
- **Area:** The need to increase the capabilities of MPSoCs leads their components to occupy more space. However, the technological trend is to develop more powerful, smaller devices. Thus, a critical research and development objective is to keep area consumption as low as possible. Some research papers have proposed manage-

ment schemes that include energy consumption, throughput, latency, and scalability thinking in area consumption.

Table 2 classifies the management research papers from the past two decades based on their optimization metrics. The table shows the metrics in order of research paper count, from the highest to the lowest, from left to right. At the end of Table 2, two additional columns are presented to highlight the research trend focus. The first identifies an NoC-based approach that recognizes papers that have addressed network-related topics, while the second identifies a self-awareness approach that recognizes papers that have mentioned some self-related properties explicitly (self-x properties).

After analyzing Table 2, we examined different time periods to identify the research trends related to the number of papers on metrics and design paradigms within the MPSoCs. Our study considered all research in recent years, narrowing the range from the last twenty years to the last five years in five-year intervals. The results are presented in Table 3.

According to our study, power and temperature are the main concerns with the highest percentage of research papers, even though works aimed at improving characteristics and solving related problems have slightly decreased over the years. The research trend for managing other metrics in MPSoCs has had its ups and downs but remains a reference for research in the field. For example, metrics such as fault tolerance have been trending almost entirely upward since they are closely related to overall system reliability and performance, or security, which has also gained importance due to the increased vulnerability of new systems to potential attacks.

On another note, NoC-focused research accounted for more than 50% of the papers analyzed, highlighting the importance of this paradigm as a communication infrastructure for MPSoCs. For this reason, various metrics appearing in our classification are closely related to NoC issues. One of these NoC-related metrics is throughput, which has declined, but it is still a major issue as NoC capacity remains a crucial issue. Latency is another NoC-related metric that has remained a research topic due to new application system requirements that demand specific deadlines for information exchange capacity within the NoC.

Finally, a research topic that has become relevant in the MPSoCs field is self-x properties. The upward research trend in self-x features, with almost 50% of research papers investigating these features in the last five years, reflects the need for systems to become self-aware. Research topics associated with this trend focus on adding different characteristics to the system to manage and perform processes without third-party intervention.

In the following subsections, we discuss the importance of NoC management in MPSoCs and present a classification of those research papers that specifically present improvements in NoC management. We also present a classification of those research papers that introduce specific awareness incorporating some self-x properties.











### 3.1.1. NoC Management Improvements in MPSoCs

An NoC is a packet-switching network using routers to interconnect the processing elements inside an MPSoC, as shown in Figure 3. It is a conjunction of micronetworks enabling communication between the processing elements, each including a network interface. The network management implicit in an NoC is fundamental to ensuring efficient and reliable performance of the communication infrastructure of an MPSoC. NoCs architecture adds parallelism to the information flow [29], which, in conjunction with multiple processing elements, allows MPSoCs to run various types of applications [9]. This makes the control and management of resources, such as task allocation and coordination of the communication infrastructure, critical to the performance and power consumption of the system. Although the NoC paradigm allows its functionality to be widely scalable and flexible, adding simplicity and modularity to the MPSoC design by decoupling communication and computation [2,23], it is also true that it faces significant challenges with the shrinking trends of its components, especially in terms of reliability and power consumption [27].

**Table 2.** Classification of management research papers according to their optimization objective or improvement.

Year	Management Goal or Improvement										
	Power Efficiency	Thermal	Latency	Fault-Tolerance	Throughput	Security	QoS	Execution Time	Area	NoC Focused	Self-x Properties
2001	[152]	[152]									
2003	[122]	[122]	[153]		[153]					[153]	
2004	[60]	[60,99]			[60,131]					[99,131]	
2005	[100,154]	[64]			[63]					[63]	[100]
2006	[123]				[56]		[56]			[56]	
2007	[155,156]	[124,155]	[66]	[137]	[115]		[115]	[66]	[66]	[66,115,137,156]	[115,137]
2008	[16,58,157]	[59]	[19]	[158]						[19,58,158]	
2009	[67,159]	[67]	[160]							[160]	[67]
2010	[161,162]	[31,125,161]	[71]	[25,101]	[125,161,162]					[71,161,162]	[25,31,71,101]
2011	[70,103]	[70,103]	[116,163]	[69]	[163]		[102,116,138]			[102,116,163]	[70]
2012	[43]	[142]	[73]	[72]	[142]					[43,73,142]	[43,72]
2013	[75,164]	[165,166]	[104,164,166]	[74]	[74,164–166]		[164]	[24,74,76]	[165]	[76,104,164–166]	[74,75]
2014	[167]	[167]								[167]	[167]
2015	[79,106]	[77,79,143]	[106,168,169]	[26,120,168,169]	[78,168]		[79]	[78,170]		[77,106,143,168–170]	[26,78,79,120]
2016	[22,44,82,127,171]	[22,126]	[82,172]	[57]	[107]	[45,139]		[81,127]	[44]	[22,44,45,82,107,139,172]	[57,107]
2017	[32,34,46,86,118,173]	[86]	[32–34,118,140]	[30,34,117] [86,121,134]	[34,86,118]	[34,84]	[86]	[85]	[33,174]	[32–34,121,134] [140,173,174]	[30,85,117,121]
2018	[4,92,175–177] [89,90]	[89,92,147,176]	[4,87,176,177]	[47,48,177–179]	[4]	[1,88]	[87,91,92]	[108]		[1,4,87,89,176] [91,147,179]	[1,4,89,177,178] [91]
2019	[38,40,51,52,180] [9]	[146,181]	[29,49,50,180,182]	[141,183]	[180]	[37,39,110,135]	[29,109]	[40,53]	[9]	[37,50,141,146,180] [9,39,52,110,135,181] [49,182,183]	[38,50,51,109,146] [29,181]
2020	[136,184–187] [20,114,145,188]	[96,112,145,187,189] [20,148]	[10,14,55,95,190]	[14,96,113,119,149]	[10,148,188,189]	[21,94,149]	[14]		[188,191]	[21,94,95,149,191] [10,14,148,185,188–190]	[21,94,119,136,191] [14,113,148]
2021	[11,54,98,192]	[98,193]	[54,192,194]	[15,195]	[54]	[41,196]	[197]			[41,54,192–196]	[11,15,195,197]
2022	[128,198,199]	[111]	[61]	[128]		[198]		[61,128,199]	[198]	[61,111,128,198,199]	[61,111,128,198]
2023	[129,130,200]	[129,201]	[62]	[130,202]						[62,201]	[62,129,130,201,202]

Table 3. Analysis of research trends according to Table 2.

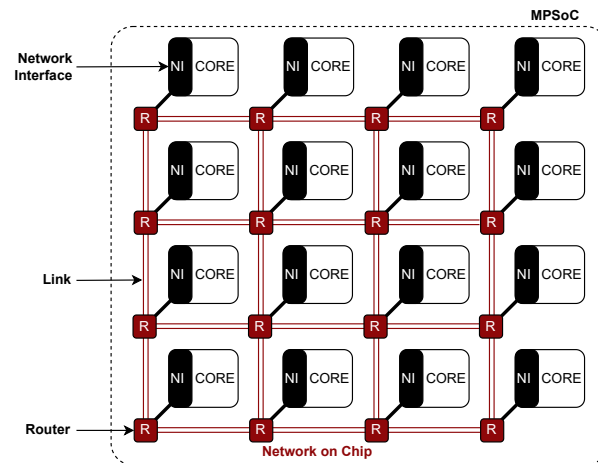
Management Goal or Improvement	Last 20 Years		Last 15 Years		Last 10 Years		Last 5 Years		Trend
	Number of Papers	Percentage	Number of Papers	Percentage	Number of Papers	Percentage	Number of Papers	Percentage	
Power efficiency	66	37.93%	55	36.67%	46	38.02%	25	37.88%	
Thermal	42	24.14%	34	22.67%	25	20.66%	14	21.21%	
Latency	40	22.99%	37	24.67%	29	23.97%	15	22.73%	
Fault-Tolerance	35	20.11%	33	22.00%	28	23.14%	12	18.18%	
Throughput	28	16.09%	22	14.67%	13	10.74%	6	9.09%	
Security	16	9.20%	16	10.67%	16	13.22%	10	15.15%	
QoS	15	8.62%	13	8.67%	9	7.44%	4	6.06%	
Execution time	15	8.62%	14	9.33%	11	9.09%	5	7.58%	
Area	9	5.17%	8	5.33%	7	5.79%	4	6.06%	
NoC Focused	97	55.75%	85	56.67%	70	57.85%	40	60.61%	
Self-x properties	58	33.33%	55	36.67%	45	37.19%	28	42.42%	

Note: The percentage of each metric considers the total number of related papers within the specified time range. There are research papers related to more than one metric.

NoCs adopt many of the concepts of traditional networks, so their management is based on a conventional network architecture consisting of three main planes: data transmission, control, and management. Basically, the control plane integrates the decision-making processes regarding the exchange of information between processing and storage elements based on established protocols, i.e., it controls the functionality of data transmission plane entities such as routers, switches, and interfaces. On the other hand, the management plane allows monitoring and configuring of the control functionality through software services [2].

NoC has become the communication infrastructure of choice for MPSoCs due to the capabilities and advantages it offers. In this context, the NoC management has been gaining importance in recent years, as shown in Table 2. NoC-related research papers are focused on improving one or more of the NoC management features like routing algorithm, network topology, buffer utilization, buffer fluidity, etc., where these improvements are aided to enhance some of the system optimization metrics. Table 4 shows a classification of the NoC-related papers according to their main optimization metric and the three most common specific NoC management improvement areas in accordance with our investigation: routing algorithm, topology, and buffer.





**Figure 3.** Typical Network-on-Chip architecture.

**Table 4.** Classification of the NoC-related papers according to their main optimization metric and the most common specific NoC management improvement area.

NoC Management Improvement Area	Management Goal or Improvement								
	Power Efficiency	Thermal	Latency	Fault-Tolerance	Throughput	Security	QoS	Execution Time	Area
Routing algorithm	[161,180,188]	[165,189,193] [161,166]	[10,50,62] [49,168,180] [104,165,166] [71,153,160] [169,182,194]	[134,168,179] [169,195]	[10,188,189] [107,168,180] [115,161,166] [153]	[21,37,41]	[115]		[165,188]
Topology	[4,44,199] [22,172,173,192]	[22,77]	[4,116,160] [169,182,190,192]	[169,183,203]	[4,107]		[116]	[170,199]	[44]
Buffer	[52,198]		[10]		[10,115]	[198]	[115]		[174,198]

- **Routing algorithm:** In an NoC, a routing algorithm is a procedure whose main objective is to forward and distribute packets from source to destination through the best path available in the MPSoC [194,204]. The related works are commonly aimed at solving the usual routing protocol problems, such as deadlock, livelock, congestion, or network faults [204]. Some of these works implement modern techniques to deal with these problems, for example, by using adaptive routing to find the shortest path and preventing possible changes in the network [194], or in other cases, by using self-properties to find a path within a faulty network [195].
- **Topology:** An NoC topology represents the physical and logical distribution of the channels and nodes within the network, and, normally, its design has a cost-performance impact in the NoC [160]. The most common NoC topologies are mesh, torus, tree, polygon, and butterfly [190]. In this context, researchers have worked in developing new topologies or modifying existing ones to implement communication infrastructure improvements like circulant topology [203] and Butterfly-Fat-Tree topology [183] for improving fault-tolerance, honeycomb topology [160] for improving network-cost, WK-Recursive topology [192] for improving power efficiency and latency, RicoBit topology [190] for improving latency, or Spidergon topology [205] for improving structure and modularity. Also, new development includes not only 2D topologies but also 3D topologies [142,143,161,173,174,189,190].
- **Buffer:** NoCs use buffers to store transmitted packets for a short period of time within a router before they are processed to be forwarded. Some works have focused on improving certain aspects related to buffering, such as prioritizing flits forwarding through buffer fluidity levels awareness [10] or reducing underutilized buffers through new buffer design and switches' operation monitoring [198].

### 3.1.2. Specific Awareness in MPSoCs

The awareness integration within the MPSoCs field is one of the most recent challenges, so many researchers have implemented specific awareness to help improve the performance of these systems. In a general definition, self-awareness alludes to an entity that is capable of being aware of its state, condition, situation, and environment [144,206]. In this context, we refer to specific awareness to the partial application of the term self-awareness in MPSoCs, i.e., that the system only knows very specific things. Although the research focused on specific awareness is far from the ideal conceptualization of whole-system self-awareness, these works have conformed a necessary precedent to identify the path toward self-aware systems. Table 5 presents the classification of the research papers implementing specific awareness. The table shows, from left to right, the type of specific awareness with the highest number of research papers to the one with the lowest number. Ultimately, we also present an extra column that identifies papers focusing on NoCs.

The purpose of Table 5 is to show the number of papers dedicated to investigating awareness within the MPSoCs. Likewise, this table helps us identify the specific types of awareness studied and their intended purpose. Table 5 is closely related to Table 3 since we can observe that the most significant number of papers have been directed to the system to focus awareness on aspects such as temperature and energy. Researchers focused about 50% of these papers on NoCs-related issues. In the NoCs context, much of this specific awareness involves managing network resources, such as traffic-aware, network-congestion-aware, network-contention-aware, workload-aware, buffer-fluidity-aware, and loss-aware (optical networks) systems. We also found papers focused on adding other types of awareness related to different aspects of the system, such as reliability, the kind of application executed, environmental fluctuations, and QoS.

- **Thermal-aware:** Thermal-aware research is concerned with implementing techniques focused on the system not exceeding the set temperature limits while dealing with its constraints and varying processes and workloads. In addition, they involve addressing challenges immersed in temperature behavior management techniques that are related, for example, with limitations on the number of sensors that can be included in the system or with the performance impact of continuous monitoring of the temperature distribution across the chip [148].
- **Energy-aware:** Since one of the main goals of modern systems is to maximize battery lifetime, researchers have aimed to improve the power performance of MPSoCs. One problem is predicting the application's behavior for adequate energy management, either by implementing known techniques or by generating new and improved ones. Consequently, some research papers have included a methodology in which the system monitors and acts on energy consumption, allowing it to improve several aspects. For example, through learning policies, the system can better respond to dynamic changes in applications [186] and to NoC processes that impact energy consumption the most [52]. Another way is by monitoring the strategies of other techniques, such as task replication, which, while improving system reliability, can also increase energy consumption too much [184].
- **Reliability-aware:** Within the MPSoC environment, reliability is related with the system's ability to respond to possible failures, so the more prepared it is to resolve failures, the more reliable it becomes. Although MPSoCs are exposed to different types of faults (see Section 3.1—Fault-tolerance), research has identified three main types that affect the reliability of electronics: manufacturing defects, constant random failures, and failures due to aging of materials [197]. As a result, monitoring the system's reliability is necessary, which consists of adequately managing the MPSoC resources, i.e., keeping the system aware of the communication infrastructure, application processes (allocation and execution of tasks), and memory performance. In this way, a reliability-aware system constantly acts at different levels to ensure specific QoS requirements.



- **Traffic-aware:** Traffic-aware research focuses on monitoring the amount of information exchanged through the communication infrastructure, usually an NoC (communication through routers). This runtime monitoring can be focused on specific key regions or distributed across the NoC. Traffic awareness allows the innovation and implementation of techniques applied in different communication processes, such as arbitration mechanisms that improve network latency [95] or routing algorithms that increase throughput [189].
- **Congestion-aware:** The congestion of the communication infrastructure of an MPSoC depends on several factors, which, in the case of NoCs, is closely related to the amount and type of traffic, latency, and network throughput. In addition, the characteristics and properties of routing and arbitration schemes play an important role in network congestion. Therefore, monitoring various metrics can improve network performance, such as leveraging information from buffers, which allows dealing with dynamic traffic loads through cognitive processes and control techniques [10]. Another improvement is identifying data flows that congest the network in certain areas or situations and subsequently avoiding them, resulting in considerable energy savings [82].
- **Environment-aware:** Environment-aware research explores the interaction between hardware and software components at different system levels and then implements management improvements with diverse objectives [68,133].
- **Application-aware:** Most NoC designs within MPSoCs do not consider the types of applications and their requirements [168]. This situation can degrade the performance of the entire system. Therefore, some papers have proposed strategies that involve application awareness at the network level, for example, by identifying the optimization metrics to which they are most sensitive and then classifying and treating them accordingly [164]. Another solution is monitoring their communication patterns and balancing the traffic load between resources by estimating routing demands [168]. In other cases, implementing continuous learning of application profiles allows the system to apply preventive and corrective actions to aid with QoS management [29].
- **Workload-aware:** The tasks of the application(s), running at any given time, define an MPSoC's workload, making it a highly variable parameter. Generally, the NoC of the MPSoC reflects the implications resulting from workload variability, since if the NoC is unaware of these variations, it may fail to manage its resources. Therefore, workload awareness is highly beneficial and can be applied to improve network performance. For example, it can enhance routing algorithms by evenly distributing NoC traffic among active resources [179]. It can also help self-recover systems from failures by identifying free processing elements at a particular time [30] or the unpredictability of runtime workload by aiding dynamic memory management [81].
- **Contention-aware:** Contention-aware research involves the system being aware of the competition in the NoC to perform intercommunication between processing elements. Given the large number of processing elements in MPSoCs, there are more concurrent parallel intercommunications, so if there is no contention-free access scheme, contentions can degrade NoC performance. Consequently, considering network contentions can help achieve different optimization objectives. This type of specific awareness can be achieved through task mapping and scheduling in communication channels [106,163], and, likewise, in optical NoCs leveraging the flexibility of adaptive routing schemes [193].
- **QoS-aware:** QoS-aware research aims to provide information that helps appropriately manage available resources to meet the application requirements. This type of specific awareness can be implemented, for example, to achieve coordinated management involving the QoS of multiple resources within a class-of-service-based architecture [138]. Similarly, QoS monitoring allows for self-adaptive QoS management at runtime, providing better resource understanding and a reactive and proactive decision-making capability [29].

- **Loss-aware:** In optical NoCs, light signals usually suffer losses while propagating through the waveguides. This condition usually requires higher power injection into the laser to counteract these losses and avoid transmission errors. Generally, the power setting of transmission lasers does not consider these losses, so a system adding the awareness of them can increase communication and energy efficiency through adaptive runtime power setting [185].
- **Fluidity-aware:** Fluidity awareness refers to understanding the fluidity in the NoCs router buffers. Researchers implement active buffer monitoring to approximate the flit fluidity levels, which helps to improve flow and congestion control [10]. A flit is the smallest entity into which information exchanged over the network is divided. In addition, fluidity awareness allows for flow prioritization, which in turn allows for better management of network resources and prediction of dynamic traffic behavior.

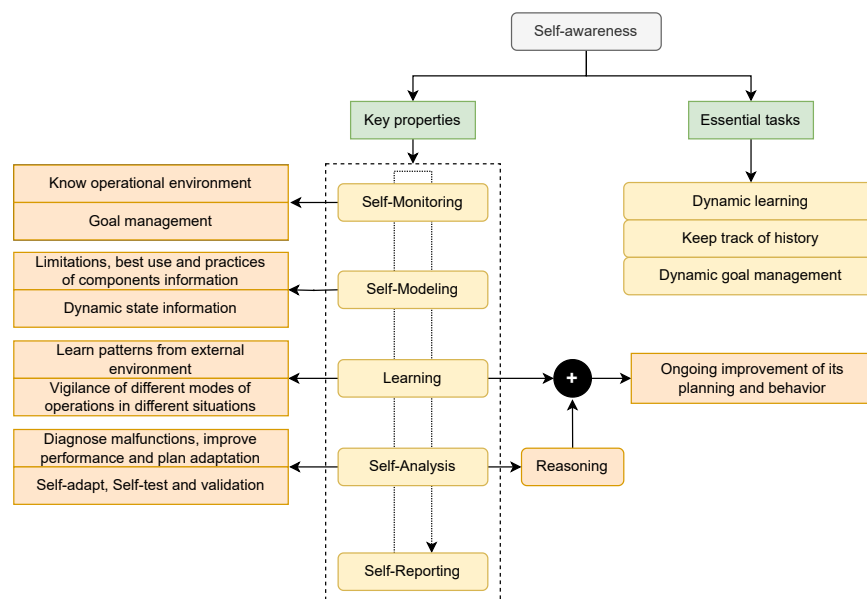
#### 4. MPSoCs, Self-Awareness, and Cyber-Physical Systems

The fusion of MPSoC with the state-of-the-art concepts of self-awareness and cyber-physical systems represents the evolution of traditional MPSoCs towards platforms that incorporate highly autonomous and self-adaptive management [12,13]. Combining these concepts within the SoC field allows us to assimilate a system capable of managing and adapting its autonomy by learning from its runtime environment. In the following subsections, we present and describe the concepts of self-awareness and cyber-physical systems.

##### 4.1. Self-Awareness

The term self-awareness is used in many fields of science and is broadly concerned with an entity being aware of its own state, condition, situation, and environment [144,206]. In 2013, as an important precedent, Kornaros et al. [17] surveyed research on intelligent systems through dynamic monitoring and management techniques. In their work, they also establish the characteristics that this type of system should have. These characteristics are proactive management and monitoring since they allow decisions at runtime based on such evaluations and make the system capable of adapting in real-time. They mention that online monitoring is the fundamental tool for a system to have adaptive runtime management. They predicted that the features of new MPSoCs had to include monitoring platforms with reconfiguration capabilities and programmability of their components.

In the last decade, although some researchers have tried to define and introduce the concept of self-awareness in the MPSoC field, many researchers have applied the concept partially. Thus, as Jantsch et al. [206] and Dutt et al. [209] mentioned in their work, it was necessary to lay the foundations of what it implies and understand its scope and benefits. The concept of self-awareness in computational systems involves not only proactive monitoring that provides information on the current state of the system and self-adaptability but also having an awareness of the model of the static and dynamic properties of the system, and thereby making decisions that trigger actions in the direction of the operation objectives [144,206]. Thus, a self-aware system can automatically adapt to changing environmental conditions and demands to meet its goals by constantly modifying its behavior and updating its components and resources [144]. Self-aware systems are intended to continuously perform a series of actions. They learn operation patterns based on different system situations and use reasoning to make decisions based on self-analysis at runtime. This is achieved by being aware of the hardware infrastructure and software architecture. Bellman et al. [12] defined the following terms as key properties of a self-aware system: self-monitoring, self-modeling, learning, self-analysis, and self-reporting (Figure 4). In addition, three essential tasks stand out from a self-aware system: dynamic learning, dynamic goal management, and keeping track of history [206].



**Figure 4.** Key properties and tasks of a self-aware system based on information presented in [12,206].

A system that integrates self-awareness is a system whose behavior is based on a constant, updated, and detailed monitoring of its own state, learning and reasoning from the interaction with its environment, and acting according to the specific objectives of the system. Therefore, self-awareness is a feature that can help the system better manage and understand its behavior, which invariably improves the use of available resources, resulting in greater efficiency [206].

#### 4.2. Cyber-Physical Systems

It is impossible to separate physical and computational processes in a computational system, as what happens in both affects each other. Thus, the computational and network entities continuously control and monitor the physical processes. The integration of computational and physical processes is represented by cyber-physical systems (CPS) [210]. These systems constantly interact with their physical environment. They must deal with aspects such as material degradation and aging, considering the constraints of their internal resources, such as computational and memory capacity [12].

##### Cyber-Physical Systems-on-Chip

The Cyber-Physical System-on-Chip (CPSoC) concept incorporates the cyber-physical systems paradigm into the SoCs field. While the design of a traditional MPSoC does not specify the explicit, monitored, coordinated, and controlled relationship of computation and communication operations with the physical environment, a CPSoC architecture incorporates an entity in charge of control, communication, and computation which interacts with the physical processes at runtime [79,107]. In addition, the structured architecture of a CPSoC allows the system to monitor different aspects through the different layers, providing essential information to deal with process variabilities. This information adds adaptability to the system, as it can be used in mechanisms capable of acting at various levels [144].

Sarma et al. [211] defined the base architecture of a CPSoC (Figure 5), where they divide it into several abstraction layers interacting with a platform composed of different sensors and actuators, whose objective is to provide the control and management of the cyber-information and the physical environment of the chip. They achieve this by using the Observe–Decide–Act (ODA) paradigm in combination with adaptive and reflective middleware that includes adaptive NoCs and some degree of self-awareness. A CPSoC platform provides a computing framework that enables the simultaneous control and management of data processing and physical environment manifestations. Thus, physical



and virtual sensors and actuators ensure data reliability by considering aspects such as power, temperature, degradation, and system performance. Sarma et al. [211] mentioned that adaptability and self-awareness can be added to each abstraction layer through these physical and virtual sensors and actuators (a combination of software and hardware).

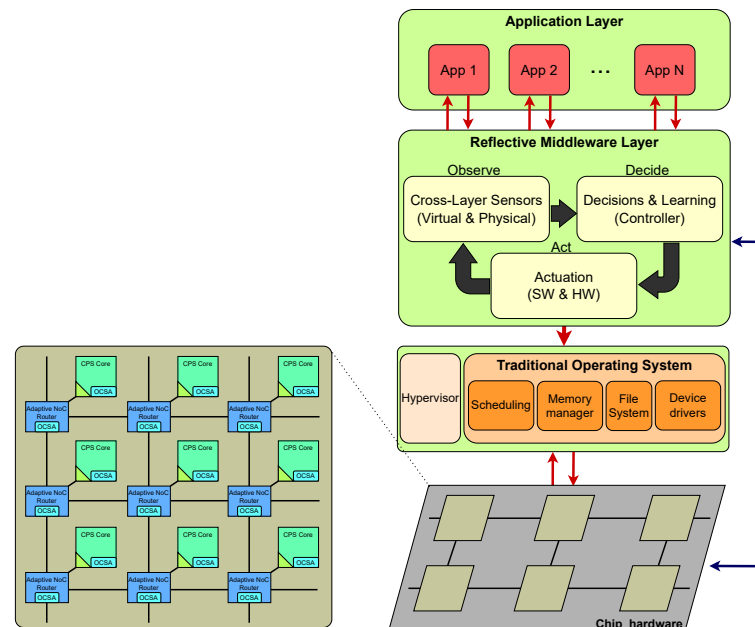


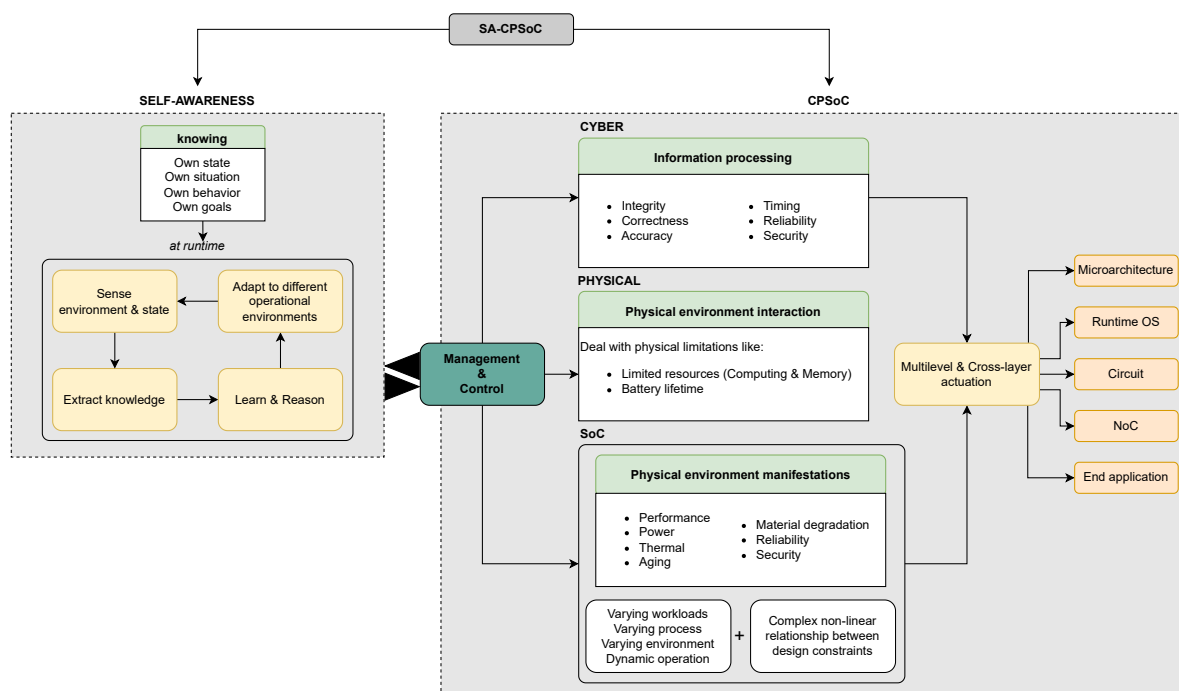
Figure 5. CPSoC architecture [211].

#### 4.3. Self-Aware Cyber-Physical Systems-on-Chip

MPSoC design has moved toward submicron platforms, with increased complexity and design requirements. These platforms integrate many processing elements into increasingly heterogeneous systems for higher functionality and performance. New applications demand increased capabilities from MPSoCs, so computation and intercommunication between their components must be faster and more efficient [79]. They must also maintain acceptable optimization metrics such as power, temperature, and energy. Thus, new MP-SoCs must be systems that constantly deal with variable processes and dynamic runtime objectives while maintaining high reliability, security, and efficiency [212]. In this way, self-aware Cyber-Physical Systems-on-Chip (SA-CPSoCs) represent a suitable solution to these demands, being CPSoCs which add self-awareness. These characteristics make an adaptive and dynamic system possible, aware of its condition, state, behavior, and what is happening in real time in its physical environment, all with little or no human intervention [12,79,144,210,213]. Thus, the design of the SA-CPSoCs allows a significant increase in adaptability through highly autonomous and intelligent system management.

The research of Bellman et al. [12] is one of the most recent works on self-aware cyber-physical systems. It defines them as self-managing systems that know their state, situation, behavior, and goals through knowledge extraction from their physical and virtual environment. These systems are supposed to learn and reason at runtime to subsequently make fast and effective adaptive decisions autonomously in the face of unexpected events. Thus, the addition of these characteristics within a system-on-chip leads to SA-CPSoCs. In this way, SA-CPSoCs increase management and control capabilities and represent the evolution of MPSoCs by adding learning and reasoning mechanisms that allow the system to self-model based on the continuous understanding of its static and dynamic properties to anticipate and correct faults. In their research, Dutt et al. [144], Jantsch et al. [206], and Bellman et al. [12] described the key properties and characteristics that CPSoCs that aim to add self-awareness must meet, considering the development and implementation challenges that this task implies (Figure 6). In addition to the self-awareness features, an SA-CPSoC monitors the behavior of different variables between the abstraction layers of

the system, using these data to implement statistical prediction and learning models. These models are used by the actuation mechanisms that perform adaptations at different system levels, such as in the intrachip communication system or in the operating system. These operations must be performed with awareness of the information processing, physical manifestations, and updated system objectives. The properties and characteristics of SA-CPSoCs aim to improve the system's autonomy, making it capable of self-managing its resources and enhancing its utilization at runtime.



**Figure 6.** SA-CPSoC characteristics, based on information presented in [12,144,206].

### Prospects, Future Development, and Challenges of SA-CPSoCs

Emerging MPSoCs need to deal with increasingly heterogeneous systems and hostile environments. In their conception, SA-CPSoCs prove to offer the capabilities required by modern and future applications where the system is required to have full control and fresh information of all its resources to act accordingly at runtime. The accelerated technological progress and its necessities force the development of tools and systems with greater capabilities, and the field of MPSoCs is no exception. The progress made in recent years in the agreement of the definition of self-awareness in this field has laid certain foundations for the development of such systems. As mentioned by Bellman et al. [12] in their work, the application of self-awareness in its entirety may not be the most profitable for all cases, and some applications may only require some of its characteristics. It is, therefore, necessary to think in the future about the design of SA-CPSoCs as a generalized design that can be applied to a wide range of applications, rather than thinking about adding self-awareness to an individual system [12]. In this context, these systems must provide a flexible infrastructure that allows for the adoption and organization of processes inherent to a self-awareness nature. However, there are several challenges to overcome to make SA-CPSoCs possible in fullness of their definition, especially in this resource-constrained system. Table 6 shows the challenges identified by some researchers who have worked the most in this area.

**Table 6.** Challenges facing the development of SA-CPSoCs [12,206].

Self-Awareness [206]	Challenges		
	What Is Needed?	SA-CPSoC [12]	What Is Needed?
Dynamic Learning	Better machine learning algorithms based on feedback signals.	Considering self-awareness, subjectivity, and situatedness.	Techniques that consider the system's own perspective in different possible situations in addition to the environmental changes. Enhancing the decision-making process.
Scalable self-awareness	Define different levels of self-awareness for different system requirements.	Building resource-sensitive self-awareness.	Consider the resources needed to implement self-awareness and its processes at runtime.
Ensuring correctness	Validate the level of systems adaptation ensuring reliability and guarantees.	Verifying self-awareness and establishing guarantees.	Methods to implement verifications of self-awareness level from the design stage and make the system understand the guarantees during operation.
Design methodology	Change the design paradigm to let the systems be self-aware.	Developing new designs and engineering processes.	Adapt design and processes to introduce self-aware CPSs characteristics including dynamic decisions instead of predefined decisions.
Formulation goals	Define more quantitative goals like adaptability, autonomy, self-assessment, and situation assessment, and formulating mechanisms to define trade-offs.	Creating an infrastructure for self-awareness processes.	New reference architectures and design templates guided to provide a generic infrastructure that facilitates the development of SA-CPS and all of its capabilities.

#### 4.4. NoCs as Self-Aware Cyber-Physical Systems

NoCs face several design and implementation challenges in modern MPSoCs where the dynamic workloads demanded by new applications impose the necessity to place several NoCs in parallel to interconnect many entities, such as processing elements, memories, and ports, to meet system performance requirements. This workload variability involves different traffic patterns within the network, which makes NoCs unpredictable, leading to system instability if proper resource management does not exist. In addition, these factors add uncertainty at design time because it is practically impossible to know all the scenarios the system would face during its operation, which decreases the efficiency and performance of a predefined design for specific applications.

For this reason, the necessity arises to design NoCs with adaptive capabilities, allowing them to meet various important requirements such as power consumption, reliability, security, response times, and performance. These requirements must be satisfied even when the system's conditions, like temperature and voltage, vary during the execution of different processes. A cyber-physical NoC with self-aware capabilities may be the first step toward SA-CPSoCs. In this context, an NoC can apply reconfiguration actions based on up-to-date knowledge of the system status and situation (active monitoring), actions such as dynamic bandwidth adaptation, routing algorithms, arbitration policies, topology, and so on, or the application of techniques such as throttling, DVFS, or clock gating to adjust power consumption at runtime. These actions make it possible to meet performance objectives or to achieve the best possible optimization by making the necessary trade-offs following the system capabilities and constraints and offering certain guarantees [2].

#### 4.5. SDNoC as a Base Architecture in the Many-Core Era

A key component for the capable and efficient management of an MPSoC is the communication infrastructure. An MPSoC communication infrastructure aims to enable communication links between the system components, taking the information from a source to a destination entirely. This sharing of information is critical for the correct operation of the MPSoC, so the infrastructure must be sufficiently robust and guarantee the necessary communication resources for each application. If the communication infrastructure cannot meet the application resources requirements such as bandwidth, throughput, latency, traffic, waiting time, or utilization time, it can become a serious problem affecting the entire system's performance.

In the MPSoC environment, four basic communication infrastructures are commonly found: point-to-point interconnection (P2P), shared bus interconnection, crossbar switch interconnection, and NoC. Although all these infrastructures' general objective is to commu-

nicate the multiple processing elements within the system, each has different capabilities. P2P interconnection implies dedicated communication links between each pair of elements, i.e., there is only direct communication between two elements where a handshake protocol controls the traffic. Shared bus interconnection implies that all elements share a communication bus controlled by an arbiter, but there can only be one active link between two elements at a time. Crossbar switch interconnection implies a communication backbone controlled by an arbiter where there can be several communication links between several elements simultaneously, as long as there is not more than one link for the same receiver. Finally, NoC infrastructure implies a packet-switching network through interconnected routers throughout the MPSoC, enabling possible communication between all the elements of the system.

Compared to its counterparts, the advantages of an NoC make it the most feasible communication infrastructure for MPSoCs. These advantages lie primarily in flexibility, scalability, and energy efficiency [8]. The NoC communication infrastructure can make the design of MPSoCs faster and more efficient, allowing the implementation of distributed schemes. For example, the system can have multiple communication links transmitting information from different segments instead of concentrating the information in a shared bus. These features contemplate new allocation, control, and monitoring challenges.

The variability of the processes and workloads of modern systems requires the NoC to implement control of adjustments that adapt the communication resources in the best possible way. This control needs to be autonomous and at runtime, i.e., the system must identify those events that may trigger adaptation actions and make decisions about them at runtime [29]. An NoC must consider, for example, network traffic, congestion, contention, and fluidity when making adjustments. Likewise, the new MPSoCs require that these actions consider the unpredictability of their processes and add intelligence to allow them to anticipate and act accordingly [27].

These characteristics are part of SA-CPSoCs, and adding them to new systems is a titanic and challenging task to carry out in a system holistically, so they must be implemented in a modular fashion with a hierarchical organization. In this way, the problem is divided into smaller and less complex dilemmas that allow progress toward the final objective. Although the resources of an MPSoC can be very varied, we can divide them into computational and communication resources [29]. The former is regarding the data processing in the information processing and storage elements. The latter relates to sharing information through the network, i.e., data transmission and reception. Management of NoC resources is a critical and fundamental part of managing an MPSoC. The NoC is responsible for interconnecting hundreds, and even thousands, of processing and storage elements within an MPSoC through reliable and secure means, allowing correct and efficient operation according to the system requirements.

In the paradigm of cyber-physical systems, control, computation, and communication are closely related. Similarly, in the management plane of an MPSoC, the management of the NoC cannot be excluded. Therefore, their interaction must be taken into account during decision making in a self-adaptive system. We believe that managing an MPSoC based on network processes increases the control and management capabilities of the entire system. The distributed scheme of an NoC allows the system to divide problems throughout the network while communication and computing resource management are also distributed. In this context, advances in developing SDNoC architectures can help achieve self-adaptive cyber-physical systems.

#### 4.5.1. SDNoCs as a Solution

Lee [210] mentions in his work that for a system to take full advantage of a CPS's capabilities, one must think in abstractions that allow contemplation of both the dynamic physical environment and the information processing. These abstractions must be included in platforms or models that efficiently manage the physical and software processes to achieve the system objectives. In addition, as mentioned by Bellman et al. [12], introduc-

ing a self-aware system's learning and reasoning capabilities into the CPS development paradigm is challenging while keeping these capabilities relatively explicit and accessible for processing.

The challenges involved in a self-aware cyber-physical NoC infer the development of a new paradigm involving the construction of computational and communication abstractions. A layered distributed approach can help address different problems modularly, so the issues of each layer are treated independently, and changes made to a specific layer will not affect other layers' behavior. The management of network resources and its related challenges, together with the requirements of the MPSoC development trend, has inspired the research of new solutions that harmoniously combine the advances and ideas in this field. Such is the case of the motivation of SDNoCs, which, as mentioned by Gomez-Rodriguez et al. [2], is related to the management problems it can solve.

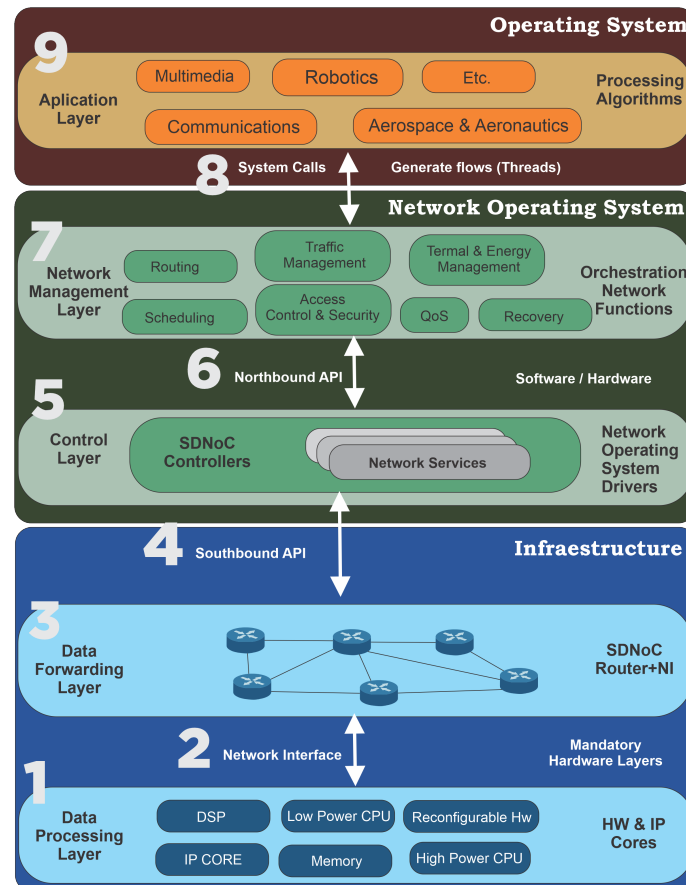
### SDNoC Architecture

Gomez-Rodriguez et al. [2] reviewed the literature and state-of-the-art for SDNoCs, clarifying their conceptualization and further explaining the initial motivation for the approach and the development path taken in recent years. Based on this work and given that the main feature of SDN is to simplify network management processes, we believe that an SDNoC architecture would allow the organization and host each of the features of an SA-CPSoC.

The term SDN dates to 1996 and arose to give the user control over network entities' data forwarding [214]. Before SDN, the main disadvantages of network systems were the interoperability between network entities from different vendors and the difficulty of implementing network configurations. Thus, the main idea of SDN was to decouple control and data planes so that the control rests out of individual network entities on a centralized controller. Within this architecture, the controller makes the forwarding decisions by having an overview of the network, and then these decisions are passed to network entities like switches to execute them [1,9]. This approach directly impacted network performance by opening the possibility for the online configuration of network entities through software-based services and for the standardization of multivendor networks through open interfaces between control and data plane devices. The SDN concept lets the system define the forwarding policy based on programmable network services, which are aware of the application [214]. Therefore, SDN emerged as a solution to simplify network management for the new intelligent applications requiring dynamic functions with reduced operational and maintenance costs [1].

More recently, Sandoval-Arechiga et al. [80] proposed introducing the SDN concept into the NoC field and leveraging its advantages within the MPSoCs. Since this proposal, various researchers started working on this concept due to its potential to increase the capabilities of the MPSoCs. This SDNoC architecture brings characteristics like programmability and abstraction capacity to the NoC management environment, opening the possibility for online reconfiguration and enabling design reuse. SDNoC architecture improves and simplifies NoC management, resulting in more system flexibility, reduced complexity of network entities (routers), real-time guarantees, and communication network self-adaptation [50]. It converts the routers into less-complex entities, which can be programmed by following a forwarding policy dictated by a centralized software controller.

The SDNoC architecture consists of three segments: application, network operating system (NOS), and infrastructure, and five layers: application, network management, control, data transmission, and data processing [2], as shown in Figure 7. This architecture has a hierarchical organization where each layer provides a service to a higher layer through a well-defined interface. In their work, Gomez-Rodriguez et al. [2] described each of this architecture's components and their possible implementations in detail.



**Figure 7.** SDNoC architecture. Odd numbers represent the layers, and even numbers represent the HW/SW interfaces. A different color represents each segment [2].

An SDNoC architecture provides certain facilities that can be exploited to implement the SA-CPSoC characteristics. Thus, SDNoC can be a valuable tool for constructing a self-awareness architecture for MPSoCs serving as the backbone of SA-CPSoCs. The layered infrastructure and abstraction allow a more straightforward connection with the monitoring and acting infrastructure. The network operating system (NOS) and the optimization machines help with the concentration of information and the definition of policies to specify more efficient processes or tasks depending on the system status and application. Communication channels and protocols are already created and ready to use, so a new service or element can be included just by adding an existing well-defined interface or controller. In this way, the SDNoC architecture facilitates the orchestration of physical and software processes through scalable administrative functions and SDNoC controllers.

A development challenge of this approach is the scalability problem using a centralized SDN controller, which can impact the performance of large-scale MPSoCs. Some researchers have proposed a distributed organization for the SDNoC controller to leverage the multiple advantages of distributed systems. Table 7 shows SDNoC research according to the controller organization and system's goal. In this way, SDNoC facilitates the communication infrastructure over a distributed system, which is one of the most critical elements of such systems. Having well-defined communications protocols for a distributed system allows new services to be set up, leveraging that backbone of communication infrastructure. For example, supposing the system requires a new thermal monitor service, it can be added by connecting to the communication infrastructure straightforwardly using standardized interfaces and just focusing on the controller design for upper layers.

From this perspective, we think that an SDNoC architecture could be an effective tool or even the main baseline for the SA-CPSoCs. There are still many details to define



and other development challenges this proposal brings, but the SDNoC architecture can potentially solve many of the future MPSoC problems.

**Table 7.** SDNoC research according to the controller organization and system’s goal.

		Management Goal or Improvement						
		General	Power Efficiency	Latency	Fault-Tolerance	Throughput	Security	QoS
Organization	Centralized	[23,35,36] [80,83]	[9,32]	[32,33,87]	[149]		[1,21,37] [41,94,149]	[87,91]
	Distributed		[4,31,44]	[14,29,50] [4]	[14]	[4]	[1,49,94]	[14,29]

## 5. Conclusions

After an extensive investigation of the state-of-the-art management within the MPSoC field over the last twenty years, this paper presents a classification of management types based on some of the issues that have driven the development of MPSoCs. The research also analyzes the optimization or improvement objectives of the research papers, identifying trends that show the importance and impact of the most exploited areas and those that are becoming increasingly relevant. Additionally, the paper identifies research papers that implement self-x properties and classifies them according to the specific type of awareness they implement to illustrate the evolution of the research and the precedents of the idea of SA-CPSoCs.

The paper describes the evolution of ideas, concepts, and developments before the conception of SA-CPSoCs as a solution to the demands of new and future MPSoCs and presents the challenges that this task implies. The paper also presents a network-based management of MPSoC that leverages the SDNoC architecture characteristics to strengthen the development of SA-CPSoCs as a conceptual idea.

The main objective of this research is to provide the scientific community with a primary point of reference in MPSoCs management and the integration of self-awareness in this field. This comprehensive and structured material will facilitate future research and developments.

**Author Contributions:** Conceptualization, G.G.-M. and R.S.-A.; formal analysis, G.G.-M., R.S.-A., L.O.S.-S., L.G.-L., S.I.-D. and J.R.S.-E.; investigation, G.G.-M., R.S.-A. and L.G.-L.; methodology, G.G.-M. and R.S.-A.; writing—original draft, G.G.-M. and R.S.-A.; writing—review and editing, G.G.-M., R.S.-A., L.O.S.-S., L.G.-L., S.I.-D., J.R.S.-E., J.R.G.-R. and V.I.R.-A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Council for the Humanities, Sciences and Technology (Consejo Nacional de Humanidades Ciencias y Tecnologías—CONAHCYT) through CVU numbers 611184, 595403 and 505017.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Acknowledgments:** We want to acknowledge and thank the support of CIDTE (Centro de Investigación, Innovación y Desarrollo en Telecomunicaciones), PITec (Posgrado en Ingeniería y Tecnología Aplicada) of UAZ and CONACYT (Consejo Nacional de Ciencia y Tecnología).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ellinidou, S.; Sharma, G.; Dricot, J.M.; Markowitch, O. A SDN solution for system-on-chip world. In Proceedings of the 2018 5th International Conference on Software Defined Systems (SDS 2018), Barcelona, Spain, 23–26 April 2018; pp. 14–19. [\[CrossRef\]](#)
2. Gomez-Rodríguez, J.; Sandoval-Arechiga, R.; Ibarra-Delgado, S.; Rodriguez-Abdala, V.I.; Vazquez-Avila, J.L.; Parra-Michel, R. A Survey of Software-Defined Networks-on-Chip: Motivations, Challenges, and Opportunities. *Micromachines* **2021**, *12*, 183. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Jeon, M.; Kim, N.; Jang, Y.; Lee, B.D. An efficient network resource management in SDN for cloud services. *Symmetry* **2020**, *12*, 1556. [\[CrossRef\]](#)

4. Scionti, A.; Mazumdar, S.; Portero, A. Towards a scalable software defined network-on-chip for next generation cloud. *Sensors* **2018**, *18*, 2330. [[CrossRef](#)] [[PubMed](#)]
5. de Dinechin, B.D.; Aygnac, R.; Beaucamps, P.E.; Couvert, P.; Ganne, B.; de Massas, P.G.; Jacquet, F.; Jones, S.; Chaisemartin, N.M.; Riss, F.; et al. A clustered manycore processor architecture for embedded and accelerated applications. In Proceedings of the 2013 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 10–12 September 2013; pp. 1–6. [[CrossRef](#)]
6. Olofsson, A. Epiphany-V: A 1024 processor 64-bit RISC System-On-Chip. *arXiv* **2016**. [[CrossRef](#)]
7. Zheng, F.; Li, H.L.; Lv, H.; Guo, F.; Xu, X.H.; Xie, X.H. Cooperative Computing Techniques for a Deeply Fused and Heterogeneous Many-Core Processor Architecture. *J. Comput. Sci. Technol.* **2015**, *30*, 145–162. [[CrossRef](#)]
8. Lee, H.G.; Chang, N.; Ogras, U.Y.; Marculescu, R. On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches. *Assoc. Comput. Mach. (ACM)* **2007**, *12*, 23. [[CrossRef](#)]
9. Nunes, F.L.D.; Kreutz, M.E. Using SDN Strategies to Improve Resource Management on a NoC. In Proceedings of the IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SoC), Cuzco, Peru, 6–9 October 2019; pp. 224–225. [[CrossRef](#)]
10. Tsai, W.C.; Chen, S.J.; Hu, Y.H.; Iun Chiang, M. Network-Cognitive Traffic Control: A Fluidity-Aware On-Chip Communication. *Electronics* **2020**, *9*, 1667. [[CrossRef](#)]
11. Dinakarrao, S.M.P. Self-aware power management for multi-core microprocessors. *Sustain. Comput. Inform. Syst.* **2021**, *29*, 100480. [[CrossRef](#)]
12. Bellman, K.; Landauer, C.; Dutt, N.; Esterle, L.; Herkersdorf, A.; Jantsch, A.; TaheriNejad, N.; Lewis, P.R.; Platzner, M.; Tammemäe, K. Self-aware Cyber-Physical Systems. *ACM Trans. Cyber-Phys. Syst.* **2020**, *4*, 38. [[CrossRef](#)]
13. Du, B.Z.; Du, B.Z.; Guo, Q.; Zhao, Y.; Zhao, Y.; Zhi, T.; Zhi, T.; Chen, Y.; Xu, Z. Self-Aware Neural Network Systems: A Survey and New Perspective. *Proc. IEEE* **2020**, *108*, 1047–1067. [[CrossRef](#)]
14. Ruaro, M.; Moraes, F.G. Multiple-objective Management based on a Distributed SDN Architecture for Many-cores. In Proceedings of the 2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI), Campinas, Brazil, 24–28 August 2020; pp. 1–6. [[CrossRef](#)]
15. Fochi, V.; Caimi, L.L.; Silva, M.H.; Moraes, F.G. System management recovery in NoC-based many-core systems. *Analog Integr. Circuits Signal Process.* **2021**, *106*, 85–98. [[CrossRef](#)]
16. Ou, J.; Prasanna, V.K. A cooperative management scheme for power efficient implementations of real-time operating systems on soft processors. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2008**, *16*, 45–56. [[CrossRef](#)]
17. Kornaros, G.; Pnevmatikatos, D. A survey and taxonomy of on-chip monitoring of multicore systems-on-chip. *ACM Trans. Des. Autom. Electron. Syst.* **2013**, *18*, 17. [[CrossRef](#)]
18. Fattah, M.; Daneshtalab, M.; Liljeberg, P.; Plosila, J. Exploration of MPSoC monitoring and management systems. In Proceedings of the 6th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC), Montpellier, France, 20–22 June 2011. [[CrossRef](#)]
19. Kim, B.; Kim, Y.; Lee, D.; Tak, S. A reconfigurable NoC platform incorporating real-time task management technique for H/W-S/W codesign of network protocols. In Proceedings of the 2008 International Symposium on Ubiquitous Multimedia Computing, Hobart, TAS, Australia, 13–15 October 2008; pp. 238–243. [[CrossRef](#)]
20. Mandal, S.K.; Ogras, U.Y.; Doppa, J.R.; Ayoub, R.Z.; Kishinevsky, M.; Pande, P.P. Online Adaptive Learning for Runtime Resource Management of Heterogeneous SoCs. *arXiv* **2020**, arXiv:2008.09728.
21. Ruaro, M.; Caimi, L.L.; Moraes, F.G. A Systemic and Secure SDN Framework for NoC-Based Many-Cores. *IEEE Access* **2020**, *8*, 105997–106008. [[CrossRef](#)]
22. Yang, L.; Liu, W.; Jiang, W.; Li, M.; Yi, J.; Sha, E.H.M. FoToNoC: A hierarchical management strategy based on folded lorus-like Network-on-Chip for dark silicon many-core systems. In Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, 25–28 January 2016; pp. 725–730. [[CrossRef](#)]
23. Berestizshevsky, K.; Even, G.; Fais, Y.; Ostrometzky, J. SDNoC: Software defined network on a chip. *Microprocess. Microsyst.* **2017**, *50*, 138–153. [[CrossRef](#)]
24. Castilhos, G.; Mandelli, M.; Madalozzo, G.; Moraes, F. Distributed resource management in NoC-based MPSoCs with dynamic cluster sizes. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Natal, Brazil, 5–7 August 2013; pp. 153–158. [[CrossRef](#)]
25. Braak, T.D.T.; Burgess, S.T.; Hurskainen, H.; Kerkhoff, H.G.; Vermeulen, B.; Zhang, X. On-line dependability enhancement of multiprocessor SoCs by resource management. In Proceedings of the 2010 International Symposium on System-on-Chip, Tampere, Finland, 29–30 September 2010; pp. 103–110. [[CrossRef](#)]
26. Wu, Z.; Fu, F.; Lu, Y.; Wang, J. A role-changeable fault-tolerant management strategy towards resilient NoC-based manycore systems. *Microelectron. J.* **2015**, *46*, 1371–1379. [[CrossRef](#)]
27. Göttinger, M.; Rahmani, A.M.; Pongratz, M.; Liljeberg, P.; Jantsch, A.; Tenhunen, H. The Role of Self-Awareness and Hierarchical Agents in Resource Management for Many-Core Systems. In Proceedings of the IEEE 10th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip (MCSoc), Lyon, France, 21–23 September 2016; pp. 53–60. [[CrossRef](#)]

28. Bragg, G.M.; Leech, C.; Balsamo, D.; Davis, J.J.; Wachter, E.; Merrett, G.V.; Constantinides, G.A.; Al-hashimi, B.M. An Application- and Platform-agnostic Control and Monitoring Framework for Multicore Systems. In Proceedings of the 8th International Joint Conference on Pervasive and Embedded Computing and Communication Systems (PECCS), Porto, Portugal, 29–30 July 2018; pp. 57–66.
29. Ruaro, M.; Jantsch, A.; Moraes, F.G. Self-adaptive QoS management of computation and communication resources in many-core SOCs. *ACM Trans. Embed. Comput. Syst.* **2019**, *18*, 37. [\[CrossRef\]](#)
30. Tsoutsouras, V.; Masouros, D.; Xydis, S.; Soudris, D. SoftRM Self-Organized Fault-Tolerant ResourceManagement for Failure Detection and Recovery in NoC Based Many-Cores. *ACM Trans. Embed. Comput. Syst.* **2017**, *16*, 144. [\[CrossRef\]](#)
31. Faruque, M.A.; Jahn, J.; Ebi, T.; Henkel, J. Runtime thermal management using software agents for multi- and many-core architectures. *IEEE Des. Test Comput.* **2010**, *27*, 58–68. [\[CrossRef\]](#)
32. Wang, J.; Feng, Q.; Wang, Y.; Dou, Q.; Dou, W. A hybrid hierarchical software-defined photonic on-chip network. In Proceedings of the 2016 International Conference on Network and Information Systems for Computers (ICNISC), Wuhan, China, 15–17 April 2016; pp. 133–137. [\[CrossRef\]](#)
33. Fathi, A.; Kia, K. A Centralized Controller as an Approach in Designing NoC. *Int. J. Mod. Educ. Comput. Sci.* **2017**, *9*, 60–67. [\[CrossRef\]](#)
34. Wachter, E.; Caimi, L.L.; Fochi, V.; Munhoz, D.; Moraes, F.G. BrNoC: A broadcast NoC for control messages in many-core systems. *Microelectron. J.* **2017**, *68*, 69–77. [\[CrossRef\]](#)
35. Delgado, S.I.; Arechiga, R.S.; Brox, M.; Ortiz, M.A. Software defined network controller: A neat solution administration for reconfigurable multi-core NoC. In Proceedings of the 2017 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 4–6 December 2018; pp. 1–4. [\[CrossRef\]](#)
36. Ellinidou, S.; Sharma, G.; Kontogiannis, S.; Markowitch, O.; Dricot, J.M.; Gogniat, G. MicroLET: A New SDNoC-Based Communication Protocol for ChipLET-Based Systems. In Proceedings of the 2019 22nd Euromicro Conference on Digital System Design (DSD), Kallithea, Greece, 28–30 August 2019; pp. 61–68. [\[CrossRef\]](#)
37. Ellinidou, S.; Sharma, G.; Rigas, T.; Vanspouwen, T.; Markowitch, O.; Dricot, J.M.; Schneider, D. SSPSoC: A Secure SDN-Based Protocol over MPSoC. *Secur. Commun. Netw.* **2019**, *2019*, 4869167. [\[CrossRef\]](#)
38. del Mestre Martins, A.L.; da Silva, A.H.L.; Rahmani, A.M.; Dutt, N.; Moraes, F.G. Hierarchical adaptive Multi-objective resource management for many-core systems. *J. Syst. Archit.* **2019**, *97*, 416–427. [\[CrossRef\]](#)
39. Madden, K.; Harkin, J.; McDaid, L.; Nugent, C. Adding Security to Networks-on-Chip using Neural Networks. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2019; pp. 1299–1306. [\[CrossRef\]](#)
40. Venkataramani, V.; Chan, M.C.; Mitra, T. Scratchpad-memory management for multi-threaded applications on many-core architectures. *ACM Trans. Embed. Comput. Syst.* **2019**, *18*, 10. [\[CrossRef\]](#)
41. Sharma, G.; Bousdras, G.; Ellinidou, S.; Markowitch, O.; Dricot, J.M.; Milojevic, D. Exploring the security landscape: NoC-based MPSoC to Cloud-of-Chips. *Microprocess. Microsyst.* **2021**, *84*, 103963. [\[CrossRef\]](#)
42. Kobbe, S.; Bauer, L.; Lohmann, D.; Schröder-Preikschat, W.; Henkel, J. DistRM: Distributed resource management for on-chip many-core systems. In Proceedings of the 2011 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Taipei, Taiwan, 9–14 October 2011; pp. 119–128. [\[CrossRef\]](#)
43. Jafri, S.M.; Guang, L.; Jantsch, A.; Paul, K.; Hemani, A.; Tenhunen, H. Self-adaptive NoC power management with dual-level agents: Architecture and implementation. In Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems (PECCS), Rome, Italy, 24–26 February 2012; pp. 450–458. [\[CrossRef\]](#)
44. Scionti, A.; Mazumdar, S.; Portero, A. Software defined Network-on-Chip for scalable CMPs. In Proceedings of the 2016 International Conference on High Performance Computing and Simulation (HPCS), Innsbruck, Austria, 18–22 July 2016; pp. 112–115. [\[CrossRef\]](#)
45. Sepulveda, J.; Flórez, D.; Immler, V.; Gogniat, G.; Sigl, G. Hierarchical group-key management for NoC-based MPSoCs protection. *J. Integr. Circuits Syst.* **2016**, *11*, 38–48. [\[CrossRef\]](#)
46. Martins, A.L.; Sant’Ana, A.C.; Moraes, F.G. Runtime energy management for many-core systems. In Proceedings of the 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS), Monte Carlo, Monaco, 11–14 December 2016; pp. 380–383. [\[CrossRef\]](#)
47. Fochi, V.; Caimi, L.L.; Silva, M.H.D.; Moraes, F.G. Fault-Tolerance at the Management Level in Many-Core Systems. In Proceedings of the 2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI), Bento Gonçalves, Brazil, 27–31 August 2018; pp. 1–6. [\[CrossRef\]](#)
48. Domingues, A.R.; Hamerski, J.C.; Amory, A. Broker Fault Recovery for a Multiprocessor System-an-Chip Middleware. In Proceedings of the 2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI), Bento Gonçalves, Brazil, 27–31 August 2018; pp. 1–6. [\[CrossRef\]](#)
49. Umoh, I.J.; Marufat, G.O.; Basira, Y.; Abdulfatai, A.D.; Muyideen, M.O. BANM: A Distributed Network Manager Framework for Software Defined Network-On-Chip (SDNoC). *Covenant J. Inform. Commun. Technol.* **2019**, *7*, 54–65.
50. Ruaro, M.; Velloso, N.; Jantsch, A.; Moraes, F.G. Distributed SDN architecture for NoC-based many-core SoCs. In Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip (NOCs), New York, NY, USA, 17–18 October 2019; pp. 1–8. [\[CrossRef\]](#)

51. del Mestre Martins, A.L.; Garibotti, R.; Dutt, N.; Moraes, F.G. The power impact of hardware and software actuators on self-adaptable many-core systems. *J. Syst. Archit.* **2019**, *97*, 42–53. [\[CrossRef\]](#)
52. Fettes, Q.; Clark, M.; Bunesco, R.; Karanth, A.; Louri, A. Dynamic Voltage and Frequency Scaling in NoCs with Supervised and Reinforcement Learning Techniques. *Computer* **2019**, *52*, 4–5. [\[CrossRef\]](#)
53. Gregorek, D.; Rust, J.; Garcia-Ortiz, A. DRACON: A Dedicated Hardware Infrastructure for Scalable Run-Time Management on Many-Core Systems. *IEEE Access* **2019**, *7*, 121931–121948. [\[CrossRef\]](#)
54. Penna, P.H.; Souto, J.V.; Uller, J.F.; Castro, M.; Freitas, H.; Méhaut, J.F. Inter-kernel communication facility of a distributed operating system for NoC-based lightweight manycores. *J. Parallel Distrib. Comput.* **2021**, *154*, 1–15. [\[CrossRef\]](#)
55. Ruaro, M.; Sant’ana, A.; Jantsch, A.; Moraes, F.G. Modular and Distributed Management of Many-Core SoCs. *ACM Trans. Comput. Syst.* **2020**, *38*, 1. [\[CrossRef\]](#)
56. Evain, S.; Diguët, J.P.; Houzet, D. NoC design flow for TDMA and QoS management in a GALS context. *Eurasip J. Embed. Syst.* **2006**, *2006*, 063656. [\[CrossRef\]](#)
57. Dutt, N.; Kurdahi, F.J.; Ernst, R.; Herkersdorf, A. Conquering MPSoC complexity with principles of a self-aware information processing factory. In Proceedings of the 2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Pittsburgh, PA, USA, 1–7 October 2016; pp. 1–4. [\[CrossRef\]](#)
58. Beigné, E.; Clermidy, F.; Durupt, J.; Lhermet, H.; Miermont, S.; Thonnart, Y.; Xuan, T.T.; Valentian, A.; Varreau, D.; Vivet, P. An asynchronous power aware and adaptive NoC based circuit. In Proceedings of the 2008 IEEE Symposium on VLSI Circuits, Honolulu, HI, USA, 18–20 June 2008; pp. 190–191. [\[CrossRef\]](#)
59. Yeo, I.; Liu, C.C.; Kim, E.J. Predictive dynamic thermal management for multicore systems. In Proceedings of the 2008 45th ACM/IEEE Design Automation Conference, Anaheim, CA, USA, 8–13 June 2008; pp. 734–739. [\[CrossRef\]](#)
60. Powell, M.D.; Goma, M.; Vijaykumar, T.N. Heat-and-run: Leveraging SMT and CMP to manage power density through the operating system. *Oper. Syst. Rev. (ACM)* **2004**, *38*, 260–270. [\[CrossRef\]](#)
61. Dalzotto, A.E.; da Silva Borges, C.; Ruaro, M.; Moraes, F.G. Non-intrusive Monitoring Framework for NoC-based Many-Cores. In Proceedings of the 2022 XII Brazilian Symposium on Computing Systems Engineering (SBESC), Fortaleza, Brazil, 21–24 November 2022; pp. 1–7. [\[CrossRef\]](#)
62. Balakrishnan, M.T.; Venkatesh, T.; Bhaskar, A.V. Design and implementation of congestion aware router for network-on-chip. *Integration* **2023**, *88*, 43–57. [\[CrossRef\]](#)
63. Avasare, P.; Nollet, V.; y Mignolet, J.; Verkest, D.; Corporaal, H. Centralized End-to-End Flow Control in a Best-Effort Network-on-Chip. In Proceedings of the 5th ACM international conference on Embedded software, Jersey City, NJ, USA, 18–22 September 2005; pp. 17–20. [\[CrossRef\]](#)
64. Merkel, A.; Weissel, A. Event-Driven Thermal Management in SMP Systems. In Proceedings of the Second Workshop on Temperature-Aware Computer Systems (TACS’05), Madison, WI, USA, 4–8 June 2005; pp. 1–10.
65. Nollet, V.; Marescaux, T.; Avasare, P.; Verkest, D.; Mignolet, J.Y. Centralized run-time resource management in a network-on-chip containing reconfigurable hardware tiles. In Proceedings of the Design, Automation and Test in Europe, Munich, Germany, 7–11 March 2005; pp. 234–239. [\[CrossRef\]](#)
66. Brand, J.W.V.D.; Ciordas, C.; Goossens, K.; Basten, T. Congestion-controlled best-effort communication for networks-on-chip. In Proceedings of the 2007 Design, Automation and Test in Europe Conference and Exhibition, Nice, France, 16–20 April 2007; pp. 948–953. [\[CrossRef\]](#)
67. Wang, Y.; Ma, K.; Wang, X. Temperature-constrained power control for chip multiprocessors with online model estimation. *ACM SIGARCH Comput. Archit. News* **2009**, *37*, 314–324. [\[CrossRef\]](#)
68. Cho, S.; Demetriades, S. MAESTRO: Orchestrating predictive resource management in future multicore systems. In Proceedings of the 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), San Diego, CA, USA, 6–9 June 2011; pp. 1–8. [\[CrossRef\]](#)
69. Braak, T.D.T.; Toersche, H.A.; Kokkeler, A.B.; Smit, G.J. Adaptive resource allocation for streaming applications. In Proceedings of the 2011 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, Samos, Greece, 18–21 July 2011; pp. 388–395. [\[CrossRef\]](#)
70. Wang, X.; Ma, K.; Wang, Y. Adaptive power control with online model estimation for chip multiprocessors. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *22*, 1681–1696. [\[CrossRef\]](#)
71. Tedesco, L.P.; Rosa, T.; Clermidy, F.; Calazans, N.; Moraes, F.G. Implementation and evaluation of a congestion aware routing algorithm for networks-on-chip. In Proceedings of the 23rd Symposium on Integrated Circuits and System Design, São Paulo, Brazil, 6–9 September 2010; pp. 91–96. [\[CrossRef\]](#)
72. Meloni, P.; Tuveri, G.; Raffo, L.; Cannella, E.; Stefanov, T.; Derin, O.; Fiorin, L.; Sami, M. System adaptivity and fault-tolerance in NoC-based MPSoCs: The MADNESS project approach. In Proceedings of the 2012 15th Euromicro Conference on Digital System Design, Cesme, Turkey, 5–8 September 2012; pp. 517–524. [\[CrossRef\]](#)
73. Kornaros, G.; Pnevmatikatos, D. Real-time monitoring of multicore SoCs through specialized hardware agents on NoC network interfaces. In Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, Shanghai, China, 21–25 May 2012; pp. 248–255. [\[CrossRef\]](#)
74. Bolchini, C.; Carminati, M.; Miele, A. Self-adaptive fault tolerance in multi-/many-core systems. *J. Electron. Test. Theory Appl. (JETTA)* **2013**, *29*, 159–175. [\[CrossRef\]](#)



75. Hoffmann, H.; Maggio, M.; Santambrogio, M.D.; Leva, A.; Agarwal, A. A generalized software framework for accurate and efficient management of performance goals. In Proceedings of the 2013 International Conference on Embedded Software (EMSOFT), Montreal, QC, Canada, 29 September–4 October 2013; pp. 1–10. [\[CrossRef\]](#)
76. Gorski, P.; Timmermann, D. Centralized traffic monitoring for online-resizable clusters in Networks-on-Chip. In Proceedings of the 2013 8th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), Darmstadt, Germany, 10–12 July 2013; pp. 1–8. [\[CrossRef\]](#)
77. Gorski, P.; Wegner, T.; Timmermann, D. Centralized and Software-Based Run-Time Traffic Management Inside Configurable Regions of Interest in Mesh-Based Networks-on-Chip. In Proceedings of the 11th International Symposium on Applied Reconfigurable Computing (ARC), Bochum, Germany, 13–17 April 2015; pp. 179–190. [\[CrossRef\]](#)
78. Paul, J.; Oechslein, B.; Erhardt, C.; Schedel, J.; Kröhnert, M.; Lohmann, D.; Stechele, W.; Asfour, T.; Schröder-Preikschat, W. Self-adaptive corner detection on MPSoC through resource-aware programming. *J. Syst. Archit.* **2015**, *61*, 520–530. [\[CrossRef\]](#)
79. Sarma, S.; Dutt, N.; Gupta, P.; Venkatasubramanian, N.; Nicolau, A. CyberPhysical-System-On-Chip (CPSoC): A self-aware MPSoC paradigm with cross-layer virtual sensing and actuation. In Proceedings of the 2015 Design, Automation and Test in Europe Conference and Exhibition (DATE), Grenoble, France, 9–13 March 2015; pp. 625–628. [\[CrossRef\]](#)
80. Sandoval-Arechiga, R.; Parra-Michel, R.; Vazquez-Avila, J.L.; Flores-Troncoso, J.; Ibarra-Delgado, S. Software defined networks-on-chip for multi/many-core systems: A performance evaluation. In Proceedings of the 2016 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Santa Clara, CA, USA, 17–18 March 2016; pp. 129–130. [\[CrossRef\]](#)
81. Tajik, H.; Donyanavard, B.; Dutt, N.; Jahn, J.; Henkel, J. SPMPool: Runtime SPM management for memory-intensive applications in embedded many-cores. *ACM Trans. Embed. Comput. Syst.* **2016**, *16*, 25. [\[CrossRef\]](#)
82. Escamilla, J.V.; Flich, J.; Casu, M.R. Increasing the Efficiency of Latency-Driven DVFS with a Smart NoC Congestion Management Strategy. In Proceedings of the 2016 IEEE 10th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip (MCSOC), Lyon, France, 21–23 September 2016; pp. 241–248. [\[CrossRef\]](#)
83. Ruaro, M.; Medina, H.M.; Moraes, F.G. SDN-Based Circuit-Switching for Many-Cores. In Proceedings of the 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Bochum, Germany, 3–5 July 2017; pp. 385–390. [\[CrossRef\]](#)
84. Caimi, L.L.; Fochi, V.; Wachter, E.; Munhoz, D.; Moraes, F.G. Secure admission and execution of applications in many-core systems. In Proceedings of the 2017 30th Symposium on Integrated Circuits and Systems Design (SBCCI), Fortaleza, Brazil, 28 August–1 September 2017; pp. 65–71. [\[CrossRef\]](#)
85. Reis, J.G.; Fröhlich, A.A. OS support for adaptive components in self-aware systems. *Oper. Syst. Rev. (ACM)* **2017**, *51*, 101–112. [\[CrossRef\]](#)
86. Rahmani, A.M.; Haghbayan, M.H.; Miele, A.; Liljeberg, P.; Jantsch, A.; Tenhunen, H. Reliability-Aware Runtime Power Management for Many-Core Systems in the Dark Silicon Era. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 427–440. [\[CrossRef\]](#)
87. Ruaro, M.; Medina, H.M.; Amory, A.M.; Moraes, F.G. Software-Defined Networking Architecture for NoC-based Many-Cores. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5. [\[CrossRef\]](#)
88. Saeed, A.; Ahmadiania, A.; Just, M. Hardware-assisted secure communication in embedded and multi-core computing systems. *Computers* **2018**, *7*, 31. [\[CrossRef\]](#)
89. Reza, M.F.; Le, T.T.; De, B.; Bayoumi, M.; Zhao, D. Neuro-NoC: Energy Optimization in Heterogeneous Many-Core NoC using Neural Networks in Dark Silicon Era. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5. [\[CrossRef\]](#)
90. Kanduri, A.; Miele, A.; Rahmani, A.M.; Liljeberg, P.; Bolchini, C.; Dutt, N. Approximation-aware coordinated power/performance management for heterogeneous multi-cores. In Proceedings of the 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 24–28 June 2018; pp. 1–6. [\[CrossRef\]](#)
91. Kostrzewa, A.; Tobuschat, S.; Ernst, R. Self-Aware Network-on-Chip Control in Real-Time Systems. *IEEE Des. Test* **2018**, *35*, 19–27. [\[CrossRef\]](#)
92. Moazzemi, K.; Kanduri, A.; Juhasz, D.; Miele, A.; Rahmani, A.M.; Liljeberg, P.; Jantsch, A.; Dutt, N. Trends in on-Chip Dynamic Resource Management. In Proceedings of the 2018 21st Euromicro Conference on Digital System Design (DSD), Prague, Czech Republic, 29–31 August 2018; pp. 62–69. [\[CrossRef\]](#)
93. Rahmani, A.M.; Jantsch, A.; Dutt, N. HDGM: Hierarchical Dynamic Goal Management for Many-Core Resource Allocation. *IEEE Embed. Syst. Lett.* **2018**, *10*, 61–64. [\[CrossRef\]](#)
94. Ruaro, M.; Caimi, L.L.; Moraes, F.G. SDN-based Secure Application Admission and Execution for Many-cores. *IEEE Access* **2020**, *8*, 177296–177306. [\[CrossRef\]](#)
95. Baharloo, M.; Khonsari, A.; Dolati, M.; Shiri, P.; Ebrahimi, M.; Rahmati, D. Traffic-aware performance optimization in Real-time wireless network on chip. *Nano Commun. Netw.* **2020**, *26*, 100321. [\[CrossRef\]](#)
96. Haghbayan, M.H.; Miele, A.; Zouv, Z.; Tenhunen, H.; Plosila, J. Thermal-Cycling-aware Dynamic Reliability Management in Many-Core System-on-Chip. In Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 1229–1234. [\[CrossRef\]](#)

97. Maurer, F.; Donyanavard, B.; Rahmani, A.M.; Dutt, N.; Herkersdorf, A. Emergent Control of MPSoC Operation by a Hierarchical Supervisor/Reinforcement Learning Approach. In Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 1562–1567. [\[CrossRef\]](#)
98. Rupanetti, D.; Salamy, H. Thermal and energy-aware utilisation management on MPSoC architectures. *Int. J. Parallel Emergent Distrib. Syst.* **2021**, *36*, 449–469. [\[CrossRef\]](#)
99. Shang, L.; Peh, L.; Kumar, A.; Jha, N. Thermal Modeling, Characterization and Management of On-Chip Networks. In Proceedings of the 37th International Symposium on Microarchitecture (MICRO-37'04), Portland, OR, USA, 4–8 December 2004; pp. 67–78. [\[CrossRef\]](#)
100. Wu, Q.; Juang, P.; Martonosi, M.; Clark, D.W. Voltage and frequency control with adaptive reaction time in multiple-clock-domain processors. In Proceedings of the 11th International Symposium on High-Performance Computer Architecture, San Francisco, CA, USA, 12–16 February 2005; pp. 178–189. [\[CrossRef\]](#)
101. Guang, L.; Nigussie, E.; Rantala, P.; Isoaho, J.; Tenhunen, H. Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip. *Trans. Embed. Comput. Syst.* **2010**, *9*, 25. [\[CrossRef\]](#)
102. Carara, E.; Almeida, G.M.; Sassatelli, G.; Moraes, F.G. Achieving composability in NoC-based MPSoCs through QoS management at software level. In Proceedings of the 2011 Design, Automation and Test in Europe, Grenoble, France, 14–18 March 2011; pp. 407–412. [\[CrossRef\]](#)
103. Kornaros, G.; Pnevmatikatos, D. Hardware-assisted dynamic power and thermal management in multi-core SoCs. In Proceedings of the 21st Edition of the Great Lakes Symposium on Great Lakes Symposium on VLSI, GLSVLSI, Lausanne, Switzerland, 2–4 May 2011; pp. 115–120. [\[CrossRef\]](#)
104. Gorski, P.; Cornelius, C.; Timmermann, D.; Kühn, V. RedNoCs: A Runtime Configurable Solution for Cluster-based and Multi-objective System Management in Networks-on-Chip. In Proceedings of the Eighth International Conference on Systems (ICONS), Seville, Spain, 27 January–1 February 2013; pp. 192–201.
105. Cemin, D.; Götz, M.; Pereira, C.E. Dynamically reconfigurable hardware/software mobile agents. *Des. Autom. Embed. Syst.* **2014**, *18*, 39–60. [\[CrossRef\]](#)
106. Han, J.J.; Lin, M.; Zhu, D.; Yang, L.T. Contention-aware energy management scheme for NoC-based multicore real-time systems. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 691–701. [\[CrossRef\]](#)
107. Sametriya, D.P.; Vasavada, N.M. HC-CPSoC: Hybrid cluster NoC topology for CPSoC. In Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 23–25 March 2016; pp. 240–243. [\[CrossRef\]](#)
108. Tsoutsouras, V.; Anagnostopoulos, I.; Masouros, D.; Soudris, D. A Hierarchical Distributed Runtime Resource Management Scheme for NoC-Based Many-Cores. *ACM Trans. Embed. Comput. Syst.* **2018**, *17*, 65. [\[CrossRef\]](#)
109. Song, Y.; Alavoine, O.; Lin, B. A self-aware resource management framework for heterogeneous multicore SoCs with diverse QoS targets. *ACM Trans. Archit. Code Optim.* **2019**, *16*, 16. [\[CrossRef\]](#)
110. Azad, S.P.; Jervan, G.; Sepulveda, J. Dynamic and Distributed Security Management for NoC Based MPSoCs. In *Lecture Notes in Computer Science, Proceedings of the ICCS 2019: 19th International Conference, Faro, Portugal, 12–14 June 2019*; Springer: Berlin/Heidelberg, Germany; pp. 649–662. [\[CrossRef\]](#)
111. Silva, A.; Weber, I.; Martins, A.L.D.M.; Moraes, F.G. Reliability Assessment of Many-Core Dynamic Thermal Management. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 27 May–1 June 2022; pp. 1590–1594. [\[CrossRef\]](#)
112. Mohammed, M.S.; Al-Kubati, A.A.; Paraman, N.; Rahman, A.A.H.A.; Marsono, M.N. Dtapo: Dynamic thermal-aware performance optimization for dark silicon many-core systems. *Electronics* **2020**, *9*, 1980. [\[CrossRef\]](#)
113. Wachter, E.W.; Kasap, S.; Zhai, X.; Ehsan, S.; McDonald-Maier, K. A Framework and Protocol for Dynamic Management of Fault Tolerant Systems in Harsh Environments. In Proceedings of the 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS), Napoli, Italy, 13–15 July 2020; pp. 1–6. [\[CrossRef\]](#)
114. Sartor, A.L.; Krishnakumar, A.; Arda, S.E.; Ogras, U.Y.; Marculescu, R. HiLITE: Hierarchical and Lightweight Imitation Learning for Power Management of Embedded SoCs. *IEEE Comput. Archit. Lett.* **2020**, *19*, 63–67. [\[CrossRef\]](#)
115. Faruque, M.A.A.; Ebi, T.; Henkel, J. Run-time adaptive on-chip communication scheme. In Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 4–8 November 2007; pp. 26–31. [\[CrossRef\]](#)
116. Motakis, A.; Kornaros, G.; Coppola, M. Dynamic resource management in modern multicore SoCs by exposing NoC services. In Proceedings of the 6th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC), Montpellier, France, 20–22 June 2011; pp. 1–7. [\[CrossRef\]](#)
117. Fochi, V.; Caimi, L.L.; Ruaro, M.; Wachter, E.; Moraes, F.G. System management recovery protocol for MPSoCs. In Proceedings of the 2017 30th IEEE International System-on-Chip Conference (SOCC), Munich, Germany, 5–8 September 2017; pp. 369–374. [\[CrossRef\]](#)
118. Kanduri, A.; Haghbayan, M.H.; Rahmani, A.M.; Liljeberg, P.; Jantsch, A.; Tenhunen, H.; Dutt, N. Accuracy-Aware Power Management for Many-Core Systems Running Error-Resilient Applications. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 2749–2762. [\[CrossRef\]](#)



119. Rambo, E.A.; Donyanavard, B.; Seo, M.; Maurer, F.; Kadeed, T.M.; Melo, C.B.D.; Maity, B.; Surhonne, A.; Herkersdorf, A.; Kurdahi, F.; et al. The Self-Aware Information Processing Factory Paradigm for Mixed-Critical Multiprocessing. *IEEE Trans. Emerg. Top. Comput.* **2020**, *10*, 250–266. [\[CrossRef\]](#)
120. Navas, B.; Sander, I.; Oberg, J. Towards cognitive reconfigurable hardware: Self-Aware learning in RTR fault-Tolerant SoCs. In Proceedings of the 2015 10th International Symposium on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC), Bremen, Germany, 29 June–1 July 2015; pp. 1–8. [\[CrossRef\]](#)
121. Azad, S.P.; Niazmand, B.; Janson, K.; George, N.; Oyeniran, A.S.; Putkaradze, T.; Kaur, A.; Raik, J.; Jervan, G.; Ubar, R.; et al. From online fault detection to fault management in Network-on-Chips: A ground-up approach. In Proceedings of the 2017 IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Dresden, Germany, 19–21 April 2017; pp. 48–53. [\[CrossRef\]](#)
122. Bellosa, F.; Weissel, A. Event-driven energy accounting for dynamic thermal management. In Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP), New Orleans, LA, USA, 27 September 2003; pp. 1–10.
123. Isci, C.; Contreras, G.; Martonosi, M. Live, runtime phase monitoring and prediction on real systems with application to dynamic power management. In Proceedings of the 2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Orlando, FL, USA, 9–13 December 2006; pp. 359–370. [\[CrossRef\]](#)
124. Coskun, A.K.; Rosing, T.S.; Whisnant, K. Temperature aware task scheduling in MPSoCs. In Proceedings of the 2007 Design, Automation and Test in Europe Conference and Exhibition, Nice, France, 16–20 April 2007; pp. 1659–1664. [\[CrossRef\]](#)
125. Zhou, X.; Yang, J.; Chrobak, M.; Zhang, Y. Performance-aware thermal management via task scheduling. *Trans. Archit. Code Optim.* **2010**, *7*, 5. [\[CrossRef\]](#)
126. Salami, B.; Noori, H.; Mehdipour, F.; Baharani, M. Physical-aware predictive dynamic thermal management of multi-core processors. *J. Parallel Distrib. Comput.* **2016**, *95*, 42–56. [\[CrossRef\]](#)
127. Ng, J.; Wang, X.; Singh, A.K.; Mak, T. Defragmentation for Efficient Runtime Resource Management in NoC-Based Many-Core Systems. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2016**, *24*, 3359–3372. [\[CrossRef\]](#)
128. Kadri, N.; Chenine, A.; Laib, Z.; Koudil, M. Reliability-aware intelligent mapping based on reinforcement learning for networks-on-chips. *J. Supercomput.* **2022**, *78*, 18153–18188. [\[CrossRef\]](#)
129. Najibi, H.; Levisse, A.; Ansaloni, G.; Zapater, M.; Vasic, M.; Atienza, D. Thermal and Voltage-Aware Performance Management of 3-D MPSoCs With Flow Cell Arrays and Integrated SC Converters. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2023**, *42*, 2–15. [\[CrossRef\]](#)
130. Haghbayan, H.; Miele, A.; Mutlu, O.; Plosila, J. Run-time Resource Management in CMPs Handling Multiple Aging Mechanisms. *IEEE Trans. Comput.* **2023**, *72*, 2872–2887. [\[CrossRef\]](#)
131. Nollet, V.; Marescaux, T.; Verkest, D. Operating-system controlled network on chip. In Proceedings of the 41st Annual Design Automation Conference, San Diego, CA, USA, 7–11 June 2004; pp. 256–259. [\[CrossRef\]](#)
132. Wisniewski, R.W.; Sweeney, P.F.; Sudeep, K.; Hauswirth, M. Performance and environment monitoring for whole-system characterization and optimization. In Proceedings of the Conference on Power/Performance Interaction with Architecture, Circuits, and Compilers, Yorktown Heights, NY, USA, October 2004; pp. 15–24.
133. Caşcaval, C.; Duesterwald, E.; Sweeney, P.F.; Wisniewski, R.W. Performance and environment monitoring for continuous program optimization. *IBM J. Res. Dev.* **2006**, *50*, 239–248. [\[CrossRef\]](#)
134. Dang, K.N.; Meyer, M.; Okuyama, Y.; Abdallah, A.B. A low-overhead soft-hard fault-tolerant architecture, design and management scheme for reliable high-performance many-core 3D-NoC systems. *J. Supercomput.* **2017**, *73*, 2705–2729. [\[CrossRef\]](#)
135. Chaves, C.G.; Azad, S.P.; Hollstein, T.; Sepúlveda, J. DoS attack detection and path collision localization in NoC-based Mpsoc architectures. *J. Low Power Electron. Appl.* **2019**, *9*, 7. [\[CrossRef\]](#)
136. Maity, B.; Donyanavard, B.; Dutt, N. Self-aware Memory Management for Emerging Energy-efficient Architectures. In Proceedings of the 2020 11th International Green and Sustainable Computing Workshops (IGSC), Pullman, WA, USA, 19–22 October 2020. [\[CrossRef\]](#)
137. Rantala, P.; Isoaho, J.; Tenhunen, H. Novel agent-based management for fault-tolerance in network-on-chip. In Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007), Lubeck, Germany, 29–31 August 2007; pp. 551–555. [\[CrossRef\]](#)
138. Li, B.; Zhao, L.; Iyer, R.; Peh, L.S.; Leddige, M.; Espig, M.; Lee, S.E.; Newell, D. CoQoS: Coordinating QoS-aware shared resources in NoC-based SoCs. *J. Parallel Distrib. Comput.* **2011**, *71*, 700–713. [\[CrossRef\]](#)
139. Reinbrecht, C.; Susin, A.; Bossuet, L.; Sepúlveda, J. Gossip NoC—Avoiding timing side-channel attacks through traffic management. In Proceedings of the 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, USA, 11–13 July 2016; pp. 601–606. [\[CrossRef\]](#)
140. Dorai, A.; Fresse, V.; Combes, C.; Bourennane, E.B.; Mtibaa, A. A collision management structure for NoC deployment on multi-FPGA. *Microprocess. Microsyst.* **2017**, *49*, 28–43. [\[CrossRef\]](#)
141. Han, K.; Lee, J.J.; Lee, W.; Lee, J. A Diagnosable Network-on-Chip for FPGA Verification of Intellectual Properties. *IEEE Des. Test* **2019**, *36*, 81–87. [\[CrossRef\]](#)
142. Rahmani, A.M.; Vaddina, K.R.; Latif, K.; Liljeberg, P.; Plosila, J.; Tenhunen, H. Generic monitoring and management infrastructure for 3D NoC-bus hybrid architectures. In Proceedings of the 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip, Lyngby, Denmark, 9–11 May 2012; pp. 177–184. [\[CrossRef\]](#)

143. Chen, K.C.J.; Chao, C.H.; Wu, A.Y.A. Thermal-Aware 3D Network-On-Chip (3D NoC) Designs: Routing Algorithms and Thermal Managements. *IEEE Circuits Syst. Mag.* **2015**, *15*, 45–69. [\[CrossRef\]](#)
144. Dutt, N.; Jantsch, A.; Sarma, S. Self-aware Cyber-Physical Systems-on-Chip. In Proceedings of the 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2–6 November 2015; pp. 46–50. [\[CrossRef\]](#)
145. Singh, A.K.; Dey, S.; McDonald-Maier, K.; Basireddy, K.R.; Merrett, G.V.; Al-Hashimi, B.M. Dynamic energy and thermal management of multi-core mobile platforms: A survey. *IEEE Des. Test* **2020**, *37*, 25–33. [\[CrossRef\]](#)
146. Said, M.; Shalaby, A.; Gebali, F. Thermal-aware network-on-chips: Single- and cross-layered approaches. *Future Gener. Comput. Syst.* **2019**, *91*, 61–85. [\[CrossRef\]](#)
147. Chittamuru, S.V.R.; Thakkar, I.G.; Pasricha, S. LIBRA: Thermal and Process Variation Aware Reliability Management in Photonic Networks-on-Chip. *IEEE Trans. Multi-Scale Comput. Syst.* **2018**, *4*, 758–772. [\[CrossRef\]](#)
148. Chen, K.C.; Tang, H.W.; Liao, Y.H.; Yang, Y.C. Temperature tracking and management with number-limited thermal sensors for thermal-aware NoC systems. *IEEE Sens. J.* **2020**, *20*, 13018–13028. [\[CrossRef\]](#)
149. Ellinidou, S.; Sharma, G.; Markowitch, O.; Gogniat, G.; Dricot, J.M. A novel Network-on-Chip security algorithm for tolerating Byzantine faults. In Proceedings of the 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati, Italy, 19–21 October 2020; pp. 1–6. [\[CrossRef\]](#)
150. Pande, P.; Grecu, C.; Ivanov, A.; Saleh, R.; Micheli, G.D. Design, Synthesis, and Test of Networks on Chips. *IEEE Des. Test Comput.* **2005**, *22*, 404–413. [\[CrossRef\]](#)
151. Bjerregaard, T.; Mahadevan, S. A survey of research and practices of network-on-chip. *ACM Comput. Surv.* **2006**, *38*, 71–121. [\[CrossRef\]](#)
152. Brooks, D.; Martonosi, M. Dynamic Thermal Management for Microprocessors. In Proceedings of the 7th International Symposium on High-Performance Computer Architecture HPCA, Monterrey, Mexico, 19–24 January 2001; pp. 171–182.
153. Nilsson, E.; Millberg, M.; Oberg, J.; Jantsch, A. Load distribution with the proximity congestion awareness in a network on chip. In Proceedings of the 2003 Design, Automation and Test in Europe Conference and Exhibition, Munich, Germany, 7 March 2003; pp. 1126–1127. [\[CrossRef\]](#)
154. Talpes, E.; Marculescu, D. Toward a multiple clock/voltage island design style for power-aware processors. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2005**, *13*, 591–603. [\[CrossRef\]](#)
155. Zhu, Y.; Albonesi, D.H. Synergistic Temperature and Energy Management in GALS Processor Architectures. In Proceedings of the 2006 International Symposium on Low Power Electronics and Design, Tegernsee, Germany, 4–6 October 2006; pp. 55–60. [\[CrossRef\]](#)
156. Leung, L.F.; Tsui, C.Y. Energy-aware synthesis of networks-on-chip implemented with voltage islands. In Proceedings of the 44th Annual Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 128–131. [\[CrossRef\]](#)
157. Kim, W.; Gupta, M.S.; yeon Wei, G.; Brooks, D. System Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators. In Proceedings of the 2008 IEEE 14th International Symposium on High Performance Computer Architecture, Salt Lake City, UT, USA, 16–20 February 2008; pp. 123–134.
158. Hosseinabady, M.; Nunez-Yanez, J. Fault-tolerant dynamically reconfigurable NoC-based SoC. In Proceedings of the 2008 International Conference on Application-Specific Systems, Architectures and Processors, Leuven, Belgium, 2–4 July 2008; pp. 31–36. [\[CrossRef\]](#)
159. Rangan, K.K.; Wei, G.Y.; Brooks, D. Thread motion: Fine-grained power management for multi-core systems. In Proceedings of the 36th International Symposium on Computer Architecture, Austin, TX, USA, 20–24 June 2009; pp. 302–313. [\[CrossRef\]](#)
160. Yin, A.W.; Xu, T.C.; Liljeberg, P.; Tenhunen, H. Explorations of Honeycomb Topologies for Network-on-Chip. In Proceedings of the 2009 Sixth IFIP International Conference on Network and Parallel Computing, Gold Coast, QLD, Australia, 19–21 October 2009; pp. 73–79. [\[CrossRef\]](#)
161. Chao, C.H.; Jheng, K.Y.; Wang, H.Y.; Wu, J.C.; Wu, A.Y. Traffic- and thermal-aware run-time thermal management scheme for 3D NoC systems. In Proceedings of the 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip, Grenoble, France, 3–6 May 2010; pp. 223–230. [\[CrossRef\]](#)
162. Tran, A.T.; Truong, D.N.; Baas, B. A Reconfigurable Source-Synchronous On-Chip Network for GALS Many-Core Platforms. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2010**, *29*, 897–910. [\[CrossRef\]](#)
163. Sinnen, O.; To, A.; Kaur, M. Contention-aware scheduling with task duplication. *J. Parallel Distrib. Comput.* **2011**, *71*, 77–86. [\[CrossRef\]](#)
164. Mishra, A.K.; Mutlu, O.; Das, C.R. A heterogeneous multiple network-on-chip design: An application-aware approach. In Proceedings of the 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 29 May–7 June 2013. [\[CrossRef\]](#)
165. Chen, K.C.; Kuo, C.C.; Hung, H.S.; Wu, A.Y.A. Traffic- and Thermal-aware Adaptive Beltway Routing for three dimensional Network-on-Chip systems. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 19–23 May 2013; pp. 1660–1663. [\[CrossRef\]](#)
166. Chao, C.H.; Chen, K.C.; Yin, T.C.; Lin, S.Y.; Wu, A.Y.A. Transport-layer-assisted routing for runtime thermal management of 3D NoC systems. *ACM Trans. Embed. Comput. Syst.* **2013**, *13*, 11. [\[CrossRef\]](#)
167. Kornaros, G.; Pnevmatikatos, D. Dynamic power and thermal management of NoC-based heterogeneous MPSoCs. *ACM Trans. Reconfigurable Technol. Syst.* **2014**, *7*, 1. [\[CrossRef\]](#)

168. Lee, D.; Parikh, R.; Bertacco, V. Highly fault-tolerant NoC routing with application-aware congestion management. In Proceedings of the 9th International Symposium on Networks-on-Chip, Vancouver, BC, Canada, 28–30 September 2015; pp. 1–8. [\[CrossRef\]](#)
169. Rahman, M.M.H.; Nor, R.M.; Sembok, T.M.B.T.; Akhand, M.A.H. Architecture and Network-on-Chip Implementation of a New Hierarchical Interconnection Network. *J. Circuits Syst. Comput.* **2015**, *24*, 1540006. [\[CrossRef\]](#)
170. Jain, A.; Kumar, A.; Sharma, S. Comparative Design and Analysis of Mesh, Torus and Ring NoC. *Procedia Comput. Sci.* **2015**, *48*, 330–337. [\[CrossRef\]](#)
171. De, V. Fine-grain power management in manycore processor and System-on-Chip (SoC) designs. In Proceedings of the 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2–6 November 2016; pp. 159–164. [\[CrossRef\]](#)
172. Ali, M.N.; Rahman, M.M.; Nor, R.M.; Sembok, T.M.B.T. A High Radix Hierarchical Interconnection Network for Network-on-Chip. In Proceedings of the 12th International Conference on Computing and Information Technology (IC2IT), Bangkok, Thailand, 8–9 July 2016; pp. 245–254. [\[CrossRef\]](#)
173. Faisal, F.A.; Rahman, M.M.; Inoguchi, Y. A new power efficient high performance interconnection network for many-core processors. *J. Parallel Distrib. Comput.* **2017**, *101*, 92–102. [\[CrossRef\]](#)
174. Fukase, N.; Miura, Y.; Watanabe, S.; Rahman, M.H. The Performance Evaluation of a 3D Torus Network Using Partial Link-Sharing Method in NoC Router Buffer. *IEICE Trans. Inf. Syst.* **2017**, *E100.D*, 2478–2492. [\[CrossRef\]](#)
175. Tarafdar, N.; Eskandari, N.; Sharma, V.; Lo, C.; Chow, P. Galapagos: A full stack approach to FPGA integration in the cloud. *IEEE Micro* **2018**, *38*, 18–24. [\[CrossRef\]](#)
176. Liu, W.; Yang, L.; Jiang, W.; Feng, L.; Guan, N.; Zhang, W.; Dutt, N. Thermal-aware task mapping on dynamically reconfigurable network-on-chip based multiprocessor system-on-chip. *IEEE Trans. Comput.* **2018**, *67*, 1818–1834. [\[CrossRef\]](#)
177. Moghaddam, M.G.; Guan, W.; Ababei, C. Dynamic Energy Optimization in Chip Multiprocessors Using Deep Neural Networks. *IEEE Trans. Multi-Scale Comput. Syst.* **2018**, *4*, 649–661. [\[CrossRef\]](#)
178. Kochte, M.A.; Wunderlich, H.J. Self-Test and Diagnosis for Self-Aware Systems. *IEEE Des. Test* **2018**, *35*, 7–18. [\[CrossRef\]](#)
179. Pano, V.; Lerner, S.; Yilmaz, I.; Lui, M.; Taskin, B. Workload-Aware Routing (WAR) for Network-on-Chip Lifetime Improvement. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5. [\[CrossRef\]](#)
180. Mehranzadeh, A.; Khademzadeh, A.; Bagherzadeh, N.; Reshadi, M. DICA: Destination intensity and congestion-aware output selection strategy for network-on-chip systems. *IET Comput. Digit. Tech.* **2019**, *13*, 335–347. [\[CrossRef\]](#)
181. Du, G.; Liu, G.; Li, Z.; Cao, Y.; Zhang, D.; Ouyang, Y.; Gao, M.; Lu, Z. SSS: Self-aware System-on-chip Using a Static-dynamic Hybrid Method. *ACM J. Emerg. Technol. Comput. Syst.* **2019**, *15*, 28. [\[CrossRef\]](#)
182. Ali, M.N.; Rahman, M.M.; Nor, R.M.; Behera, D.K.; Sembok, T.M.T.; Miura, Y.; Inoguchi, Y. SCCN: A Time-Effective Hierarchical Interconnection Network for Network-On-Chip. *Mob. Netw. Appl.* **2019**, *24*, 1255–1264. [\[CrossRef\]](#)
183. Bhanu, P.V.; Kulkarni, P.V.; Joshi, S. Butterfly-Fat-Tree topology based fault-tolerant Network-on-Chip design using particle swarm optimisation. *J. Exp. Theor. Artif. Intell.* **2019**, *31*, 781–799. [\[CrossRef\]](#)
184. Yeganeh-Khaksar, A.; Ansari, M.; Ejlali, A. ReMap: Reliability Management of Peak-Power-Aware Real-Time Embedded Systems through Task Replication. *IEEE Trans. Emerg. Top. Comput.* **2020**, *10*, 312–323. [\[CrossRef\]](#)
185. Sunny, F.; Mirza, A.; Thakkar, I.; Pasricha, S.; Nikdast, M. LoraX: Loss-aware approximations for energy-efficient silicon photonic networks-on-chip. In Proceedings of the 30th ACM Great Lakes Symposium on VLSI (GLSVLSI), Beijing, China, 7–9 September 2020; pp. 235–240. [\[CrossRef\]](#)
186. Mandal, S.K.; Bhat, G.; Doppa, J.R.; Pande, P.P.; Ogras, U.Y. An energy-aware online learning framework for resource management in heterogeneous platforms. *arXiv* **2020**, arXiv:2003.09526.
187. Pagani, S.; Manoj, P.D.; Jantsch, A.; Henkel, J. Machine Learning for Power, Energy, and Thermal Management on Multicore Processors: A Survey. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2020**, *39*, 101–116. [\[CrossRef\]](#)
188. Alaei, M.; Yazdanpanah, F. A Dynamic Congestion Management Method for Reconfigurable Network on Chip. *J. Soft Comput. Inf. Technol.* **2020**, *9*, 74–86.
189. Lee, S.C.; Han, T.H. Q-Function-Based Traffic- and Thermal-Aware Adaptive Routing for 3D Network-on-Chip. *Electronics* **2020**, *9*, 392. [\[CrossRef\]](#)
190. Satish, J.A.; Taqhi, H.; Mishra, H.; Reddy, P.C.; Sanju, V. RiCoBiT—A topology for the future multi core processor: A concept analysis and review of literature. In Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 9–10 October 2020; pp. 234–239. [\[CrossRef\]](#)
191. Bhowmik, B.; Deka, J.K.; Biswas, S. Reliability Monitoring in a Smart NoC Component. In Proceedings of the 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Glasgow, UK, 23–25 November 2020; pp. 20–23. [\[CrossRef\]](#)
192. Zhang, H.; Wang, X. KGT: An Application Mapping Algorithm Based on Kernighan-Lin Partition and Genetic Algorithm for WK-Recursive NoC Architecture. In Proceedings of the Intelligent Computing Theories and Application: 17th International Conference, ICIC 2021, Shenzhen, China, 12–15 August 2021; pp. 86–101. [\[CrossRef\]](#)
193. Li, M.; Liu, W.; Duong, L.H.; Chen, P.; Yang, L.; Xiao, C. Contention-Aware Routing for Thermal-Reliable Optical Networks-on-Chip. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2021**, *40*, 260–273. [\[CrossRef\]](#)

194. Monakhov, O.G.; Monakhova, E.A.; Romanov, A.Y.; Sukhov, A.M.; Lezhnev, E.V. Adaptive Dynamic Shortest Path Search Algorithm in Networks-on-Chip Based on Circulant Topologies. *IEEE Access* **2021**, *9*, 160836–160846. [\[CrossRef\]](#)
195. Romanov, A.; Myachin, N.; Sukhov, A. Fault-Tolerant Routing in Networks-on-Chip Using Self-Organizing Routing Algorithms. In Proceedings of the IECON 2021—47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October 2021; pp. 1–6. [\[CrossRef\]](#)
196. Chaves, C.G.; Sepúlveda, J.; Hollstein, T. Lightweight Monitoring Scheme for Flooding doS Attack Detection in Multi-Tenant MPSoCs. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 22–28 May 2021; pp. 1–5. [\[CrossRef\]](#)
197. Sahoo, S.S.; Ranjbar, B.; Kumar, A. Reliability-Aware Resource Management in Multi-/Many-Core Systems: A Perspective Paper. *J. Low Power Electron. Appl.* **2021**, *11*, 7. [\[CrossRef\]](#)
198. Singh, S.; Ravindra, J.; Naik, B.R. Proffering Secure Energy Aware Network-On-Chip (Noc) Using Incremental Cryptogine. *Sustain. Comput. Inform. Syst.* **2022**, *35*, 100682. [\[CrossRef\]](#)
199. He, J.; Xiao, Y.; Bogdan, C.; Nazarian, S.; Bogdan, P. A Design Methodology for Energy-Aware Processing in Unmanned Aerial Vehicles. *ACM Trans. Des. Autom. Electron. Syst.* **2022**, *27*, 4. [\[CrossRef\]](#)
200. Sundari, K.S.; Narmadha, R. Design energy efficient shared distributed memory management system on SoC's to improve memory performance. *Appl. Nanosci.* **2023**, *13*, 1691–1701. [\[CrossRef\]](#)
201. Ali, J.; Maqsood, T.; Khalid, N.; Madani, S.A. Communication and aging aware application mapping for multicore based edge computing servers. *Clust. Comput.* **2023**, *26*, 223–235. [\[CrossRef\]](#)
202. Cherezova, N.; Shubin, K.; Jenihhin, M.; Jutman, A. Understanding fault-tolerance vulnerabilities in advanced SoC FPGAs for critical applications. *Microelectron. Reliab.* **2023**, *146*, 115010. [\[CrossRef\]](#)
203. Sukhov, A.M.; Romanov, A.Y.; Selin, M.P. Virtual Coordinate System Based on a Circulant Topology for Routing in Networks-On-Chip. *Symmetry* **2024**, *16*, 127. [\[CrossRef\]](#)
204. Gabis, A.B.; Koudil, M. NoC routing protocols—Objective-based classification. *J. Syst. Archit.* **2016**, *66–67*, 14–32. [\[CrossRef\]](#)
205. Tatas, K.; Siozios, K.; Soudris, D.; Jantsch, A. *The Spidergon STNoC*; Springer: New York, NY, USA, 2014; pp. 161–190. [\[CrossRef\]](#)
206. Jantsch, A.; Dutt, N.; Rahmani, A.M. Self-Awareness in Systems on Chip—A Survey. *IEEE Des. Test* **2017**, *34*, 8–26. [\[CrossRef\]](#)
207. Azadi, A.; Attarzadeh-Niaki, S.H.; Shekofteh, Y. Model-Based Design of A Real-time Context-Aware Speech Enhancement System on an FPGA-SoC. In Proceedings of the 2020 20th International Symposium on Computer Architecture and Digital Systems (CADSD), Rasht, Iran, 19–20 August 2020; pp. 31–34. [\[CrossRef\]](#)
208. Zhang, Y.W.; Chen, R.K. A survey of energy-aware scheduling in mixed-criticality systems. *J. Syst. Archit.* **2022**, *127*, 102524. [\[CrossRef\]](#)
209. Dutt, N.; Jantsch, A.; Sarma, S. Toward smart embedded systems: A self-aware system-on-chip (SoC) perspective. *ACM Trans. Embed. Comput. Syst.* **2016**, *15*, 22. [\[CrossRef\]](#)
210. Lee, E.A. Cyber physical systems: Design challenges. In Proceedings of the 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, USA, 5–7 May 2008; pp. 363–369. [\[CrossRef\]](#)
211. Sarma, S.; Dutt, N.; Gupta, P.; Nicolau, A.; Venkatasubramanian, N. On-chip self-awareness using cyberphysical-systems-on-chip (CPSoC). In Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Uttar Pradesh, India, 12–17 October 2014; pp. 1–3. [\[CrossRef\]](#)
212. Sarma, S.; Dutt, N.; Gupta, P.; Venkatasubramanian, N.; Nicolau, A. *CyberPhysical-System-on-Chip (CPSoC): Sensor-Actuator Rich Self-Aware Computational Platform*; University of California Irvine: Irvine, CA, USA, 2013; pp. 1–26.
213. Göttinger, M.; Juhász, D. RoSA: A Framework for Modeling Self-Awareness in Cyber-Physical Systems. *IEEE Access* **2020**, *8*, 141373–141394. [\[CrossRef\]](#)
214. Sezer, S.; Scott-Hayward, S.; Chouhan, P.; Fraser, B.; Lake, D.; Finnegan, J.; Viljoen, N.; Miller, M.; Rao, N. Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Commun. Mag.* **2013**, *51*, 36–43. [\[CrossRef\]](#)

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.