



REVIEW

A Comprehensive Literature Review of AI-Driven Application Mapping and Scheduling Techniques for Network-on-Chip Systems

Naveed Ahmad¹, Muhammad Kaleem², Mourad Elloumi³, Muhammad Azhar Mushtaq²,
Ahlem Fatnassi⁴, Mohd Fazil⁵, Anas Bilal^{6,*} and Abdulbasit A. Darem^{7,4}

¹Department of Software Engineering, University of Sargodha, Sargodha, 40100, Punjab, Pakistan

²Department of Information Technology, University of Sargodha, Sargodha, 40100, Punjab, Pakistan

³Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, Bisha, 61922, Saudi Arabia

⁴Department of Computer Science, College of Science, Northern Border University, Arar, 73213, Saudi Arabia

⁵College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia

⁶College of Information Science and Technology, Hainan Normal University, Haikou, 571158, China

⁷Center for Scientific Research and Entrepreneurship, Northern Border University, Arar, 73213, Saudi Arabia

*Corresponding Author: Anas Bilal. Email: a.bilal19@yahoo.com

Received: 21 October 2025; Accepted: 17 December 2025; Published: 29 January 2026

ABSTRACT: Network-on-Chip (NoC) systems are progressively deployed in connecting massively parallel megacore systems in the new computing architecture. As a result, application mapping has become an important aspect of performance and scalability, as current trends require the distribution of computation across network nodes/points. In this paper, we survey a large number of mapping and scheduling techniques designed for NoC architectures. This time, we concentrated on 3D systems. We take a systematic literature review approach to analyze existing methods across static, dynamic, hybrid, and machine-learning-based approaches, alongside preliminary AI-based dynamic models in recent works. We classify them into several main aspects covering power-aware mapping, fault tolerance, load-balancing, and adaptive for dynamic workloads. Also, we assess the efficacy of each method against performance parameters, such as latency, throughput, response time, and error rate. Key challenges, including energy efficiency, real-time adaptability, and reinforcement learning integration, are highlighted as well. To the best of our knowledge, this is one of the recent reviews that identifies both traditional and AI-based algorithms for mapping over a modern NoC, and opens research challenges. Finally, we provide directions for future work toward improved adaptability and scalability via lightweight learned models and hierarchical mapping frameworks.

KEYWORDS: Application mapping; mapping techniques; network-on-chip; system on chip; optimisation

1 Introduction

Network-on-Chip (NoC) architectures have emerged as the *de facto* communication paradigm for modern multiprocessor system-on-chip (MPSoC) platforms, especially as new computing systems transition to massively parallel megacore designs. NoCs offer a structured, modular, and packet-switched communication infrastructure; they can provide an alternative for connecting high core count, heterogeneous workloads at low-power and scalable latency. Traditional shared-bus and point-to-point interconnects cannot meet the scalability, latency, and power-efficiency requirements of such systems. Consequently, application mapping



and scheduling, that is, how the computation tasks are assigned to the processing elements as well as how the communications are arranged, have become leading factors in system performance, scalability, and reliability [1,2].

The initial work with NoC application mapping concentrated on the analytical and optimization-based formations to govern interaction overhead and congestion. To avoid link saturation and reduce the average communication delay, a few solutions have proposed bandwidth-constrained mapping strategies with topology-aware placement of cores and traffic distribution [3–5]. Meanwhile, mapping heuristics focusing on energy- and performance-awareness utilized the routing flexibility and bandwidth reservation approaches to decrease communication energy consumption while satisfying latency and throughput constraints [6–8]. The key performance metrics (latency, throughput, energy efficiency) are characterized by these seminal works and are still the basis of modern NoC mapping research.

With rising system size and complexity of the NoC-based systems, these early designs introduced fault tolerance and adaptability as key design goals. Others proposed adaptive remapping techniques to dynamically recover from processing element failures to achieve a functional system with a lower overhead [3,9]. Later works expanded this idea to fault-tolerant application mapping and scheduling in the context of transient and permanent fault models, showing that reliability-aware task placement can effectively decrease performance losses in large-scale NoC implementations [10,11]. Similarly, fault-aware and QoS-aware routing approaches have been introduced to augment mapping decisions by reliably delivering packets with QoS guarantees despite link and router failures [12].

While exact optimization formulations are accurate, they are scaling-limited due to the NP-hard nature of NoC application mapping. As a result, metaheuristic optimization methods became popular as scalable alternatives. However, with their ability to escape local optimum and balance the exploration and exploitation in large design spaces, bio-inspired algorithms such as particle swarm optimization, grey wolf optimization [8,13] have been extensively utilized, which have resulted in lower communication cost, latency, and power compared to classical heuristics [14]. Evolutionary search with local refinement forms hybrid variants that significantly improve convergence behavior and robustness, and are also designed to accommodate real-time and heterogeneous NoC workloads [14–19].

Recently, research has moved more directly towards AI-based mapping and scheduling. In this context, reinforcement learning (RL) has arisen as a dominant framework for performing this sort of learning by seeking effective mapping heuristics without the necessity of large, labelled datasets. Reinforcement Learning (RL)-based frameworks have been able to consistently outperform traditional heuristic approaches by learning policy-based placement sequences that optimize both communication cost and runtime [20,21]. Mapping algorithms based on machine learning also minimize such factors as the symmetry problem of regular NoC topologies and data augmentation, and multi-label learning strategies, which further improve mapping accuracy [22,23]. This class of learning-based approaches is a significant advancement towards adaptive and scalable NoC optimization.

With the growing interest in three-dimensional (3D) NoC architectures, there are additional challenges in reducing thermal hotspots and power density. To provide reliability and stability during long-term operation, it has become imperative for vertically stacked systems to implement thermal- and traffic-aware mapping strategies. Three existing thermal-aware mapping approaches based on within-cache neural network and heuristic search have shown significant improvement on peak temperature, temperature variation, and packet latency through workload distributions among multiple dies [23–27]. In addition, recent work on application–architecture mapping for NoC-based manycore systems emphasizes energy-efficiency and high-performance objectives, reinforcing mapping as a primary lever for optimizing power–performance trade-offs in manycore designs [28]. Recently, an approach driven by reinforcement learning is proposed

to enhance the system temperature via migration of workloads to sites with varying thermal gradients for the runtime management of 3D NoC [29]. Deep learning has also allowed for further improvement in fault tolerance. Recently proposed transformer-based reinforcement learning models for fault-tolerant application mapping demonstrate significantly better communication cost, latency, and run-time performance in comparison to the prior RL and meta-heuristic baselines [30–32]. This also illustrates that deep sequence models can be used to model the spatial and temporal dependencies of NoC mapping decisions, hence building added robustness under faulty operation.

Due to the fast widening of methodologies, multiple wide-ranging surveys and reviews have categorized the systematic implementation and planning methods of NoC over static, dynamic, hybrid, and learning-based paradigms. This approach crystallizes taxonomies, quantitative comparisons, and open challenges like real-time adaptability, energy efficiency, and scalability, which are all urgent needs for management [1,33–37]. Recent works [8,38,39] provide performance evaluation frameworks and hybrid mapping platforms, which facilitate fair benchmarking and on practical deployment of such advanced NoC mapping strategies. In light of the aforementioned developments, this paper provides a comprehensive literature survey on AI-based application mapping and scheduling methods for Network-on-Chip systems, focusing on 3D NoC architectures and new learning-based dynamic models. We assign existing approaches based on optimization objectives, power efficiency, fault tolerance, load balancing, and runtime adaptability, and use core performance metrics such as latency, throughput, response time, and error rate to measure their efficiency [1,2,24,33,40,41]. Finally, we highlight various remaining research challenges and discuss potential pathways towards lightweight learned models, hierarchical mapping frameworks, and heterogeneous workload support for future NoC-based megacore systems.

Looking deeper into the architectural reasons why Network-on-Chip systems have become so prevalent, Fig. 1 provides a visual comparison of the main benefits of NoC architectures in contrast to traditional interconnection schemes. As illustrated, NoCs decouple computation from communication to use a distributed packet-switched fabric, which avoids global, centralized control as well as long global wires. It also improves the scalability, system testing, and fault isolation for system-level manycore platforms due to the architectural decoupling, by making room for systems-in-package, or systems-on-chip (SoC), integrating hundreds of cores. Additionally, its NoC architecture is flexible and can support many voltage and frequency domains for fine granularity power management and a fine ability to adapt the workload dynamically. This topology-driven design enables the seamless addition of links and processing elements without the need to halt the operation of the system while providing different levels of guarantee for latency, bandwidth, and reliability. When taken together, these properties inherently render NoC architectures as ideal candidates to exploit advanced application mapping and scheduling strategies, even more for adaptive, energy-efficient, high-quality, predictable performance in AI-driven and 3D NoC systems.

The growth of high-performance computing devices is exponential, and their performance increasingly depends on how well resources, parallel communication, and interconnects can be optimised. This trend also guides the evolution of chip manufacturing in the SoC direction, integrating multiple heterogeneous cores into one silicon platform for higher throughput and density. This combination increases the processing performance, energy efficiency, scale, and fault tolerance of computing devices [42]. With the ongoing increase of computing core counts, the NoC approach has become the communication substrate of choice to support the next generation of SoCs, capable of abstract application mapping, traffic scheduling, and adaptive load balancing in large multi-core systems.

Bus-based communication approaches may help in minimizing area when the communication is simple, but they suffer from scalability; hence, NoC has been proven as a promising way of SoC communication [43]. NoC is a packet-switched communication overhead that has a computer network-like architecture compared

to traditional buses in microchip systems. These architectures provide benefits such as better scalability, modularity, and power efficiency [44]. Nevertheless, Network-on-Chip comes with its design challenges. NoC has two key research domains: thermal-aware routing [45–48] application mapping, floor planning [49], and thermal solutions [50]. Application mapping: This refers to the partitioning of an application over nodes and determines what each node does during execution [51]. NoC is one of the critical aspects, and in one sentence, application mapping is a critical aspect of NoC. An effective application mapping will result in improvement of the complete system, and conversely [52,53].

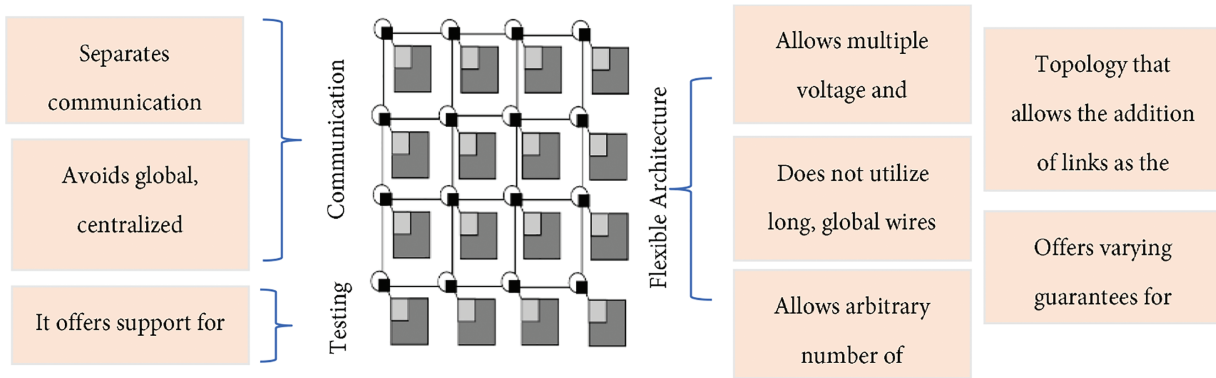


Figure 1: Advantages of network-on-chip

In NoCs, distributing multiple processing elements in a networked fashion increases the importance of application mapping in optimizing performance. Key goals include reducing inter-node communication overhead, minimizing latency, and balancing computational loads across the chip [38,54,55]. Recent research focuses on algorithms ranging from genetic algorithms [22], reinforcement learning [56,57] and neural networks [58] to address these goals. For example, deep reinforcement learning techniques have been applied to dynamically adjust task-to-node mappings based on system workload, demonstrating improved throughput and energy efficiency. Fig. 2 presents the number of documents published on application mapping across different years.

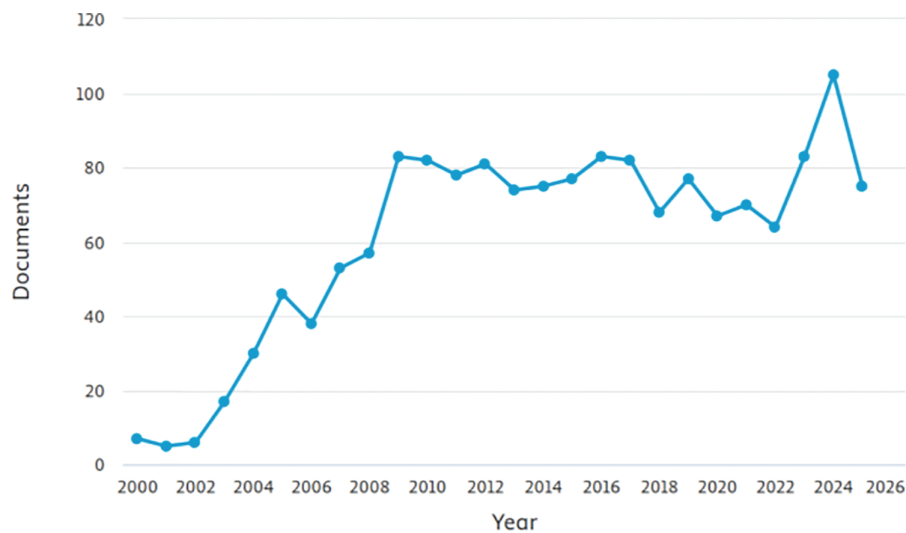


Figure 2: Documents published in application mapping in different years

As NoC architectures evolve to support dynamic 3D stacking [59] and heterogeneous cores [60], submitting the procedures of mapping the different applications becomes significantly more challenging and complex, requiring the integration of more innovative and flexible approaches to mapping. Some studies have explored the use of graph theory to model the application tasks [61–64] and their interactions, and this brought about closer mapping and better optimization of network resources. Some of the recent features that have been added to NoC include thermal mapping and the usage of thermal profiles to avoid the formation of hot regions, which affect the chips' lifespan in the system [65], but instead ensure good distribution of thermal variation throughout. Energy-aware mapping techniques, on the other hand, are another strategy that seeks to minimize power consumption in NoC while enhancing the utilization of PEs by varying the voltage and clock frequency of PEs according to workload conditions [66]. Despite these advances, current literature lacks a unified synthesis of existing approaches, particularly in how emerging techniques such as federated learning or hybrid reinforcement learning (RL) models address real-time challenges in large-scale NoC environments. Fig. 3 shows the distribution of publication types in NoC literature.

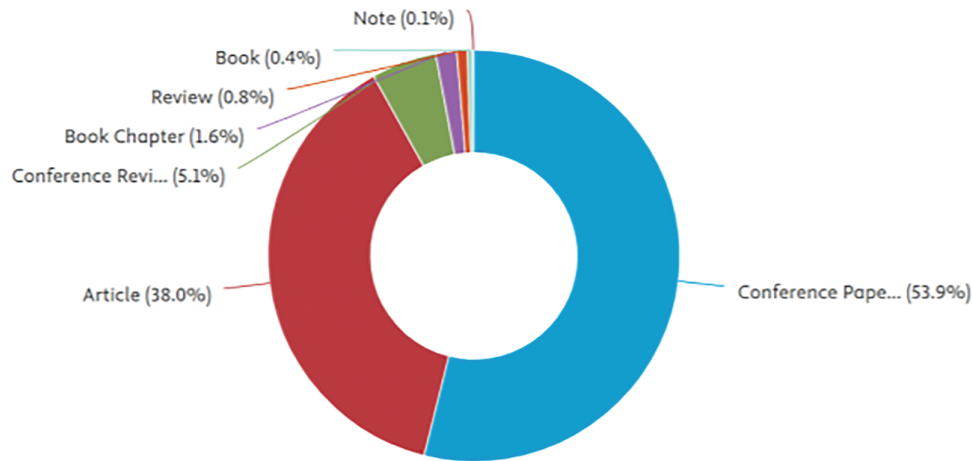


Figure 3: Distribution of publication types in NoC literature

The travel-time-based task mapping proposed by [67] for a NoC-based DNN accelerator assigns tasks are assigned unevenly to PEs based on both a fixed NoC topology (Fig. 4) and dynamically obtained congestion. They show that communication-aware mapping strategies are beneficial, as their approach improves layer-wise and full-network performance by up to 13.75 per cent over traditional row-major and distance-based mappings. The study [68] presents NoCDAS, an accelerator simulator for NoC-based DNN simulation that captures both the computation flow and the hardware microarchitecture. A mapping approach using ILP has been introduced for 3D-stacked DRAM-PIM setups [66]; this method links layer-level execution timing with underlying hardware utilisation to effectively reduce CNN latency. Unlike conventional strategies, their technique distributes the memory workload more evenly across processing units, thereby cutting the total processing time for branched CNN models.

This paper addresses that gap by presenting a focused, structured review of application-mapping methodologies tailored for modern NoC architectures. It highlights key challenges, including dynamic workload adaptation, energy efficiency, and fault tolerance. Mapping strategies are categorised into six major themes: machine learning-based mapping, power-aware techniques, dynamic approaches, fault-tolerant mechanisms, static mapping, and load-balancing schemes. The key challenges associated with application mapping in NoC [69] are analysed [70], and techniques such as RL and fault-tolerant mapping methods are evaluated. The taxonomy of application mapping techniques classifies recent work into various categories,

i.e., machine learning, power management, dynamic mapping, fault tolerance, static mapping, and load distribution. To the best of our knowledge, this is the only comprehensive review of application mapping in NoC that considers recent state-of-the-art techniques.

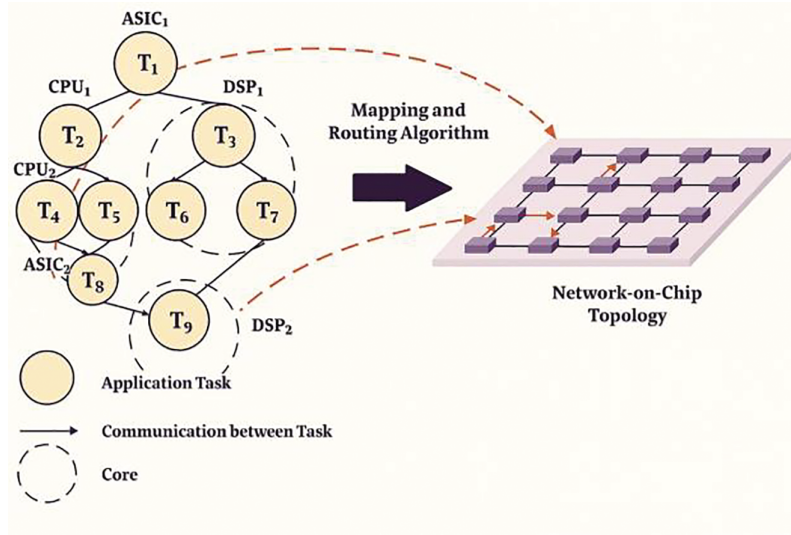


Figure 4: Network-on-chip topology

The key contributions of this paper are the following:

- Does not concentrate on the design aspects but on application mapping in NoC architectures.
- Makes comparisons and discusses the reinforcement learning and energy-aware mapping techniques as per the literature.
- Explores existing issues and possible remedies to enhance adaptability and energy amplification under dynamic workloads.
- Makes mapping methodologies consistent within a single taxonomy.
- Highlights future work on scalability, adaptability, and security, and provides a basis for future research.

The rest of the paper is structured as follows: [Section 2](#) explains the methodology for selecting the literature. [Section 3](#) discusses the challenges of application mapping in NoCs, and [Section 4](#) provides a taxonomy of recent application-mapping techniques. [Section 5](#) discusses application mapping performance metrics. Discussion and future directions are presented in [Section 6](#). [Section 7](#) is the conclusion of this study.

2 Materials and Methods

The digital libraries, protocols, and materials used in this research are highlighted. The subsections describe the search strings and parameters, the inclusion and exclusion criteria for the literature, the extraction methods, and the multi-part statistical evaluation of the selected literature before analysing the findings of the selected studies.

2.1 Search Engines and Digital Libraries

To conduct comprehensive and focused research, it is imperative to explore a wide range of digital repositories. Numerous sophisticated digital libraries are readily available online for academic research. The digital library referenced covers a broad spectrum [71] of the application mapping in NoC.

Search Strings and Timeline:

To ensure comprehensive coverage, Boolean search strings were used in various databases. The search phrases included combinations such as:

- “Network-on-Chip AND application mapping”,
- “NoC AND dynamic scheduling”,
- “Machine learning-based NoC mapping”,
- and “fault-tolerant NoC techniques”.

The publication period was restricted from 2014 to 2025, and only English-language peer-reviewed papers were considered.

Digital Libraries:

Digital libraries such as ScienceDirect, Springer, IEEE Xplore, and ACM-Digital-library were considered while collecting literature for application mapping in NoC

Search Engines:

Various academic search engines, such as Research Rabbit, Connected Paper, and Google Scholar, were checked to identify the latest available sources. The search results for primary studies are reviewed and verified in accordance with the protocol’s standards.

2.2 Primary Studies Selection

The search query is implemented within the search platform to gather information from digital repositories. The retrieved academic articles pertain to the search query specified in the research protocol. The process of acquiring pertinent data will be finalised within a fortnight. Fig. 5 shows the step-by-step procedure followed for data compilation. A total of 643 papers were obtained from various digital databases. Search and filtering were conducted in accordance with basic guidelines inspired by the PRISMA framework, aiming for transparency in identification, screening, eligibility, and inclusion. A formal PRISMA diagram is included. Fig. 6 represents the flow of data refinement across each stage. Each exclusion step was carefully logged to minimise selection bias, and duplicate entries across databases were also removed.

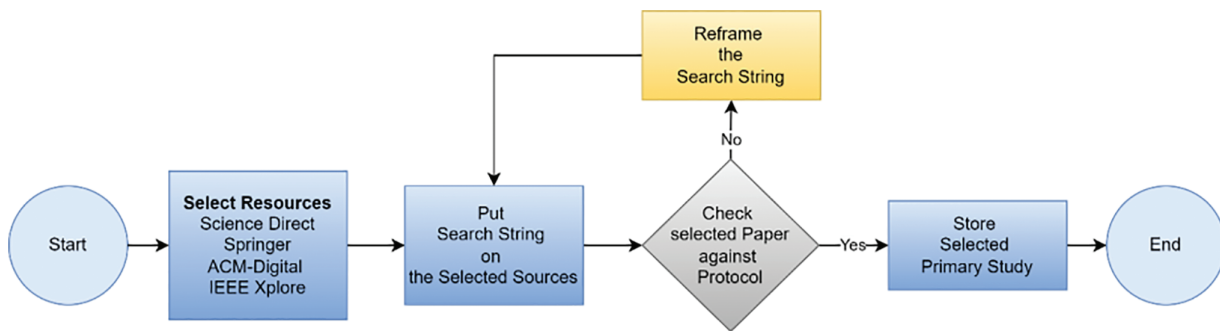


Figure 5: The procedure for data selection

Thorough scrutiny is given to both the title and abstract of every paper retrieved. After the paper evaluations, 354 research articles were identified for inclusion as primary studies (Filter 1). Fig. 6 shows the data selection after filtering.

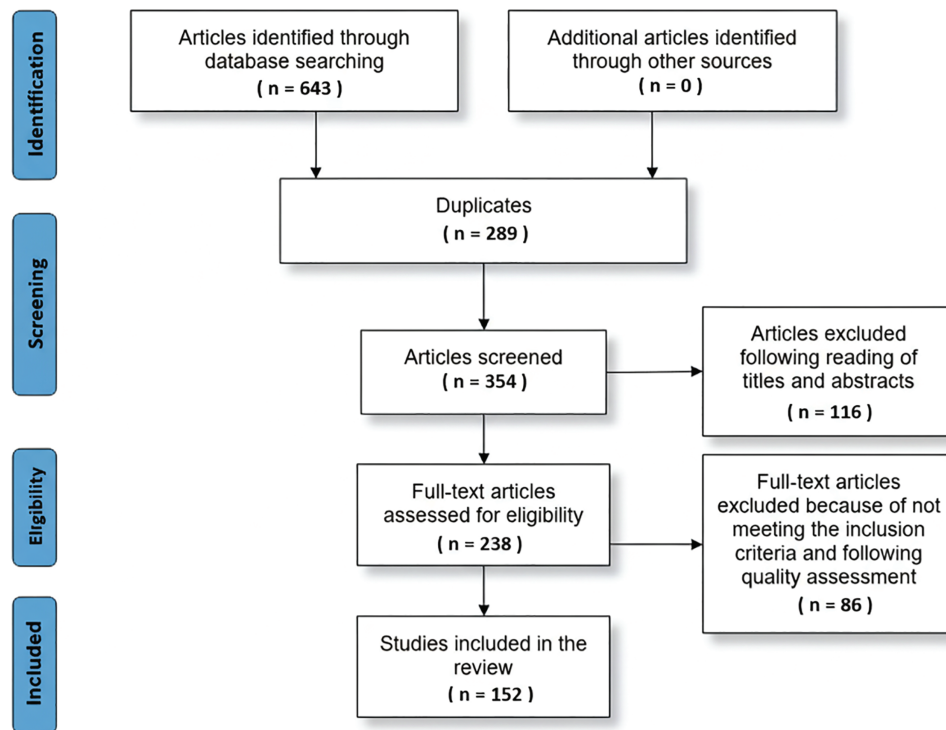


Figure 6: Primary data selection

The papers are filtered in Filter 1 and then filtered again in Filter 2. This step involved analysing the lengthy paper to determine its applicability. Many studies were disregarded in this phase because they were irrelevant. Ultimately, 238 papers were chosen following Filter 2. Information on the number of retrieved papers ultimately chosen is provided in [Table 1](#).

Table 1: Search outcomes

Sr.	Data source	Initially scanned	After filter 1	After filter 2	Final selected
1	IEEE xplore	243	123	94	70
2	Google scholar	114	93	67	63
3	ACM digital lib.	160	87	39	08
4	Science direct	89	34	26	05
5	Springer	37	17	12	06
	Total	643	354	238	152

Although only English-language, peer-reviewed academic articles were included in this study, grey literature, such as technical reports and dissertations, was briefly reviewed but excluded due to inconsistent peer-review standards and insufficient methodological details.

2.3 Scope and Exclusions

The given review focuses on the application mapping and scheduling methods of NoC systems, with particular attention to task allocation, communication optimisation, and performance evaluation across varying topologies. Other topics not related to this, such as routing algorithm design, router microarchitecture, low-level circuit optimisation, and buffer or link-level optimisation, are not covered. This category, though also connected, is a set of hardware-specific design trade-offs rather than mapping-level trade-offs, and it requires more analytical treatment than this paper's analytical capabilities allow.

3 Challenges of Application Mapping

Mapping multiple applications to various nodes is complicated. It requires a thorough understanding of all the node conditions, locations, traffic loads, workloads, thermal difficulties, network statuses, and faults. Due to this diverse nature of NoC dynamics, application mapping becomes more challenging. The key challenges associated with application mapping in NoC, as shown in Fig. 7, are adapting to varying workloads and maintaining energy efficiency. Techniques such as RL [20] and fault-tolerant mapping methods play a significant role.

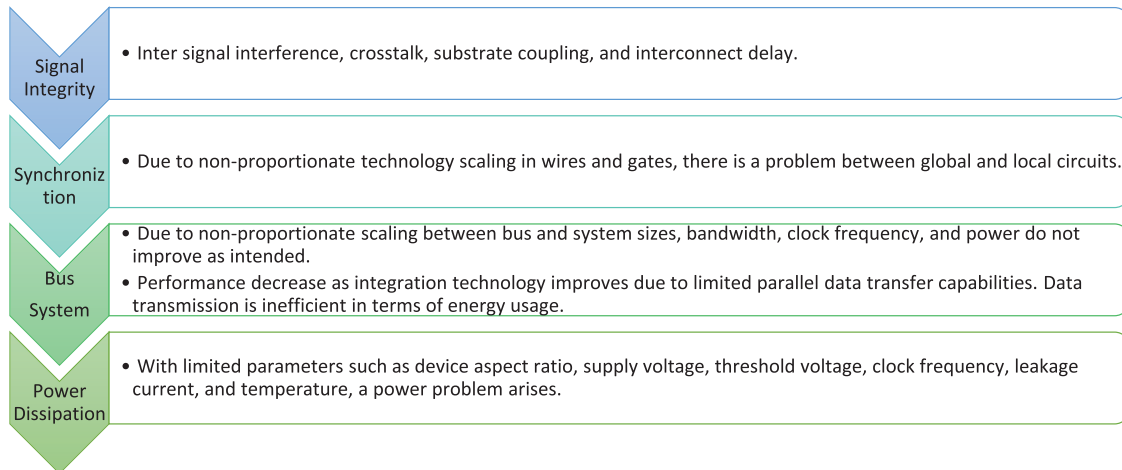


Figure 7: Challenges in SoC

SaHNoC [72] tries to balance energy efficiency and speedy communication by deciding where to put wireless switches [73]. The complexity of application mapping increases dramatically with the number of processing elements and communication links [74], making it difficult to manage communication overhead and share resources effectively. ERT-EAM tries to ensure a harmonious performance, but doing this in real-time for diverse applications while minimizing space is a real challenge [51]. Consider a scenario where a specific processing element in the network gets heated due to a heavy workload. Temperature sensors built into the chip can be used to predict when this PE would fail, which can cause the NoC to react adaptively. Moreover, Deep RL tries to configure resources dynamically according to their congestion, and this becomes a big challenge [32,75].

FANC [76], a machine learning-based approach, tries to optimize communication paths, but ensuring it works faultlessly for various scenarios is challenging. Mapping the application onto NoC is similar of finding the perfect arrangement for optimal energy usage, minimal delays, and efficient processing [13]. The machine learning model here attempts to find the optimal path for NoC communication. However, ensuring it does not provide a wrong turn, leading to performance loss, is a big challenge [22]. Designing a Ring NoC for

differently organized nodes is a challenge [77]. Evaluating 2D and 3D mesh NoC includes ensuring effective communication while considering scalability and flexibility, is really challenging [78].

4 Taxonomy of Application Mapping

The primary purpose of this study is to explore state-of-the-art methodologies, measures, tools, and algorithms for the allocation of applications on NoC architectures, shown in Fig. 8. A comprehensive description of all primary research data has been provided. Upon the conclusion of the empirical inquiries, crucial data is gathered, and deficiencies in existing research literature are identified. This work consists of scholarly articles that focus on mapping applications to loud platforms [79].

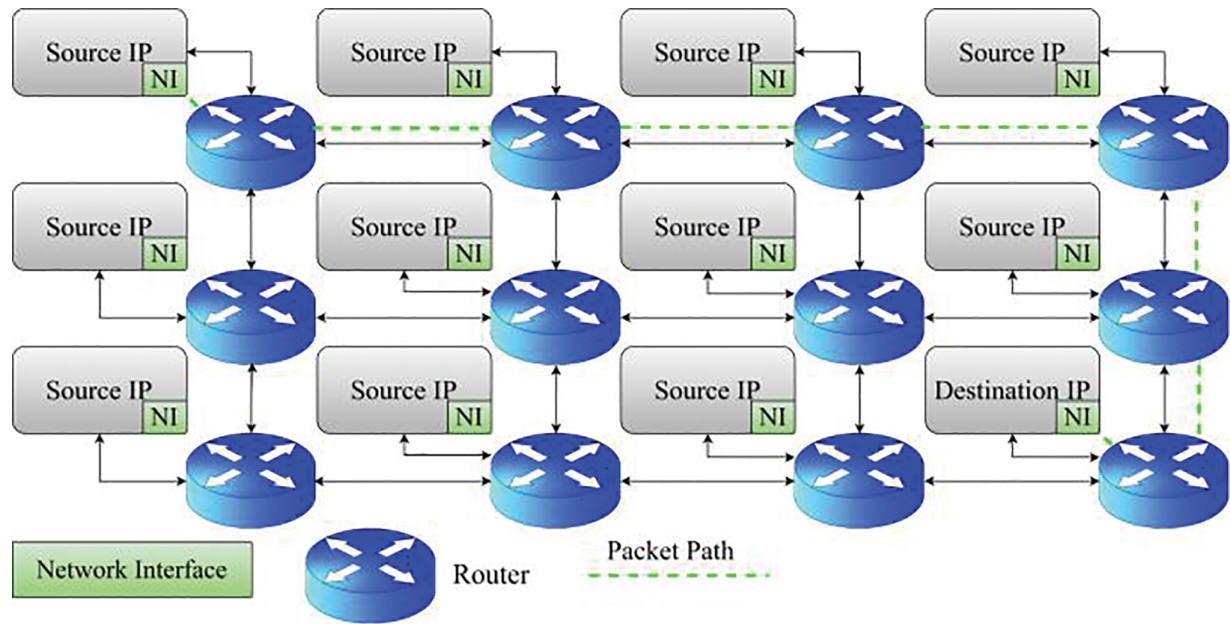


Figure 8: Conventional NoC architecture

4.1 Taxonomy Design Rationale

To systematically organise the diverse techniques for application mapping in NoC architectures, this study adopts a four-fold classification: dynamic, static, hybrid, and machine learning-based. This taxonomy reflects the temporal nature of the mapping strategy (dynamic vs. static), the integration of multiple techniques (hybrid), and the incorporation of learning-based optimisation approaches (machine learning).

Dynamic and static mapping are treated as core categories based on whether mapping decisions are made at runtime or pre-deployment. Hybrid mapping techniques [80], which combine different paradigms (e.g., GA + PSO or wire + wireless switches), often overlap with these categories but are presented separately to highlight their integrative nature. Similarly, machine learning-based approaches [69] (such as reinforcement learning) are embedded, but warrant dedicated discussion due to their growing role in optimising complex NoC environments.

This taxonomy acknowledges that overlaps exist; for instance, reinforcement learning is dynamic by nature but also represents a machine learning technique. Therefore, cross-references are made where appropriate. This structure allows for a more precise comparison of approaches within and across each category.

4.2 Categorisation of Techniques

4.2.1 Dynamic Mapping

Dynamic mapping in NoC is adaptive to changing conditions in real-time, ensuring optimal resource utilization and performance [81]. Different subcategories provide various aspects of dynamic mapping. Energy-aware mapping approach optimizes energy consumption by intelligently allocating tasks to energy-efficient resources. Recent research, such as [82] explores novel techniques to minimize energy usage [83] while maintaining system performance. Congestion-aware mapping strategies alleviate network bottlenecks by considering traffic patterns and load distribution. Work presented in [84] delve into algorithms to mitigate congestion effects for efficient mapping. Fault-aware mapping ensures system reliability by considering potential failure scenarios and distributing tasks accordingly. Recent studies, including [37], propose methodologies to enhance fault tolerance in dynamic mapping environments. Latency, execution time-based mapping approach prioritizes minimizing latency and optimizing execution time for critical applications. Recent advancements, as seen in [85] propose algorithms to dynamically adjust mappings to meet stringent latency requirements.

SaHNoC proposes a self-adaptive hybrid NoC structure that enables maximum energy efficiency in 2D mesh topologies by effectively adapting to the online environment [72]. This method innovatively chooses wireless pathways, which, in response, overcome the power consumption obstacles with wireless interconnects. With the aid of using both wired and wireless switches, SaHNoC is able to resolve the issue of communication speed, together with energy efficiency via a hybrid system.

Considering real-time applications with embedded systems, the ERT-EAM mapping strategy was introduced to improve the processing elements on the Xilinx Zynq UltraScale+ MPSoC. The ZCU104 Evaluation Kit is used as a piece of evidence [51]. This method is based on the Minimum Core Average Distance (CAD) principle [86], which is a more efficient design than the Half CoreWidth concept and also has the advantage of better latency and delay metrics, which means NoC throughput and latency.

The work presented proposes the reliability of a variety of NoC types [32], can be enhanced by a combination of reinforcement learning and multi-objective particle swarm optimization. The application modifies task allocation through RL, and additionally, it redistributes the tasks when any of the permanent process elements encounter failures. This output serves as an illustration of the value placed on reliability in modern NoC designs.

To reduce energy usage, in the study [66], authors provide a quadratic programming QP-based approach for application mapping in 2D mesh-based hybrid WiNoCs. Although it works well with small applications, its lengthy execution durations prevent it from being used with more nodes. simulated annealing SA-based approach that effectively finds optimum or nearly optimal solutions for a range of issue sizes while tackling the same mapping task. An analysis of QP and SA techniques under different wireless router locations was made, showing how router placement affects communication cost and system architecture. This study includes the quantity of jobs and routers that affect performance, showing that the SA approach regularly produces quicker, more efficient solutions than the slower QP model.

In order to navigate around the bottleneck, authors in [40] investigated the method of eliminating a few neurons during the mapping process. They introduced a method for dropping neurons depending on their degrees of contribution, which was based on ranking and selection (RSM). The system-wide and circuit-specific application of neuron ranking and selection to identify the most contributing defective neurons for mapping. A stuck-at augmented pruning (SAP) method is introduced to enhance spiking neural network defect tolerance. This method focuses on damaged neurons identified by their weight values. A technique of ranking and selection (RSM) for fault-tolerant mapping of neurons in neuromorphic hardware. The problem

of replacing many damaged neurons with fewer spare ones is resolved by this approach. The problem of certain faulty neurons not being picked for mapping is addressed by a cluster-by-cluster rating and selection method. In a system where each cluster has at least one malfunctioning neuron ($F_N = 1$), this method offers a fair selection in which each cluster is chosen for repair. Fig. 9 depicts the taxonomy of application mapping in NoC.

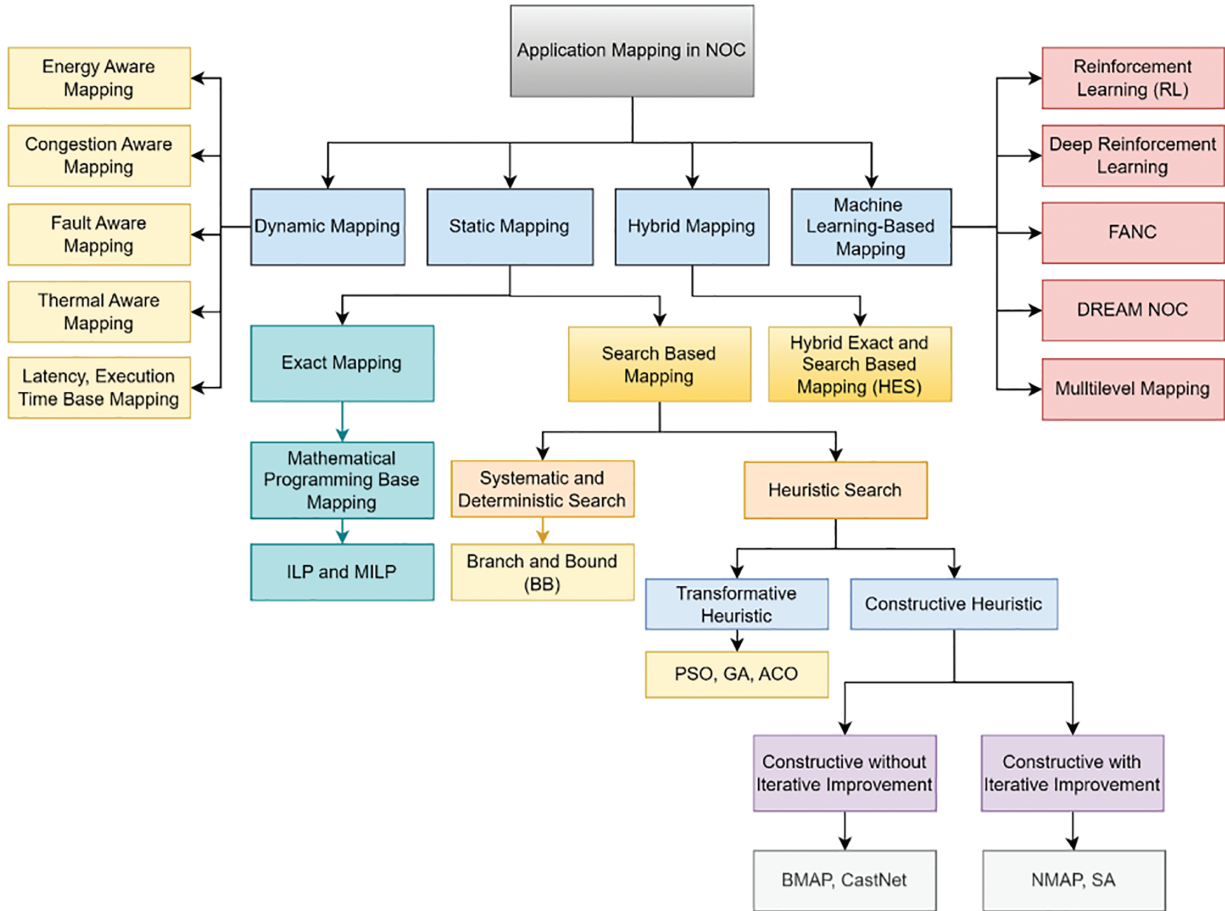


Figure 9: Taxonomy of application mapping in NoC

4.2.2 Static Mapping

Static mapping methods allocate tasks to resources based on predetermined configurations, offering stability and predictability. Exact mapping seeks the optimal allocation solution within given constraints, ensuring precise task placements. Studies such as [87] present algorithms for achieving optimal mappings in static environments. Search-based mapping employs heuristic or metaheuristic algorithms to find near-optimal solutions efficiently. Work presented in [88] explores heuristic approaches to address the complexity of mapping tasks in static environments.

The static mapping notion is investigated by digitally designing and simulating Ring NoC to suit First in First Out (FIFO) buffers in network interface cards with differently configured nodes [77]. Introducing improved performances such as more efficient data transfers and the ability to overcome bus-based communication, the work demonstrates the duty of NoCs in performing the traditional communication function.

The work presented in [78] where scalability and flexibility of 2D and 3D Mesh NoCs are brought up, just demonstrating their readiness to provide an effective bridge between different IP modules and cores. The work is evaluated in terms of area, latency, throughput, and power to reduce the processing overhead in the future.

GAPSO, a method that imprints map-related applications into the NoCs via the combination of PSO and GA [89]. Genetics is compared to PSO by the GAPSO with the intention to improve communication efficiency by using an actual Cores Graph. The suggestion to the double PSO algorithm for an energy-efficient 3D bus topology architecture [16] demonstrates attempts made to optimize the heat dissipation in 3D NoC designs. This approach in effect produces optimized heat dissipation efficiency, reduces energy loss, and improves performance at a $4 \times 4 \times 4$ 3D NoC scale.

The study is focusing on the problem of a lack of information during control transmission in NoCs because control messages are communicated as multicast and thus the available bandwidth is only partially used. The proposed solution is to have an adaptive distribution [90]. This diversity allocation approach boosts bandwidth usage, thereby indicating a solution to the problem of unproductive bandwidth of static traffic registration in NoC design.

The software and a better algorithm are introduced in this paper to identify the optimal circulant graphs with small average path length and diameter [91,92]. It presents new sets of optimal circulant graphs, offers an analytical expression of their width of the bisection, and deduces constant time relations of degree eight circulants. This study uses a genetic algorithm to improve uneven NoC layouts with mixed router types. By doing so, it cuts down the delay while keeping time-sensitive data on schedule. At the same time, space usage shrinks, boosting overall efficiency [93].

The work presented in [94], proposed a task mapping approach based on heuristics is presented, which aims to improve processor core aging and communication overhead simultaneously. Multicore heterogeneous architectures are the backbone of most cloud and edge computing systems. As a result, they have deliberately targeted multicore systems. In order to simultaneously maximize network load and lifespan reliability of multicore-based edge computing servers, a communication and aging-aware task mapping technique is presented. When mapping tasks, the presented approach operates by avoiding hotspot places. Critical applications are placed on edge computing servers perform better when workloads are mapped to alternative cores while avoiding hotspots and taking communication requirements into account.

An Artificial Bee Colony (ABC)-based technique for mapping application cores to 3D-NoC architectures is presented in [95]. In terms of the overall energy usage, the findings of the study have been compared to several multimedia benchmarks. The outcomes demonstrate that the suggested strategy beats CastNet and produces results for energy usage that are quite similar to those of ILP. Recent studies suggested the new method outperforms ILP-based models, as ILP generates optimum mappings over a very long time. Compared to an integer linear programming (ILP) based model that takes a very long time to find the optimal solution, the suggested method yields results that are equivalent in less time.

For multicore hybrid wireless NoC HWNoC-based systems, a lightweight dynamic application mapping [37] and a scheduling method is provided [96]. The suggested method makes use of long-distance wireless networks' benefits to reduce communication energy usage and assist in meeting application deadlines. The suggested method determines whether the incoming tasks may be scheduled in accordance with the core usage threshold or not. The suggested mapping and scheduling approach is compared to well-known resource allocation approaches using experimental data from a cycle-accurate HWNoC simulator. The comparison is made in terms of communication energy consumption, communication latency, deadline

violation rate, and incurred runtime overhead for various test scenarios. The experimental findings verified that, in comparison to the other alternatives, the suggested strategy is highly competitive.

In another study, a linear programming mathematical model has been presented to map jobs onto a multicore system on chip (MCSoc) [97]. The model incorporates the energy budget limitations, computer and communication capability of the MCSoc, as well as the computation and communication requirements of the applications. MCSocs based on on-chip connectivity networks are to be equipped with SA and GA algorithms for mapping activities. For mapping in energy-constrained MCSoc systems, the SA and GA are combined with the energy and performance limitations of the suggested linear programming model. Using the cycle-accurate NoC model Garnet and the Embedded System Synthesis Benchmarks Suite (E3S) [98], the SA and GA techniques were assessed on the Gem5 platform. Mapping results were generated for latency and performance measurement on the gem5/garnet platform using an in-house simulator. The results of the simulation show that, for 2D-mesh NoCs with 36–100 cores, the simulated annealing approach performs better than the genetic algorithm in a variety of applications when compared to the E3S benchmarks.

In another study [99], a technique was presented that impacts the issues arising from congestion in NoC, reliability, and efficiency of energy in multiprocessor system on chips (MPSocs). They presented a task mapping algorithm that incorporates the concept of dynamic voltage and frequency scaling to minimize energy while also aiming to distribute the workload relatively equitably across different tiles while avoiding interference between tiles communicating through the NoC-based platform, with the objectives of satisfying real-time constraints and achieving high reliability. The paper proposes a Mixed-Integer Linear Programming (MILP) approach as a method of finding the best solution and offers a 3-stage heuristic that involves task distribution, the adjustment of frequency, and the minimization of contention. Many simulations obtained show less contention, as well as a savings of energy, than the previous techniques. Specifically, to enhance the footprints in NoC-based MPSocs, the accordant study focuses on the capability of communication contention in task mapping for enhanced proficiency of the system.

In particular, Ref. [100] proposes a new branch and bound routing algorithm for optical networks-on-chip (ONocs) based on silicon photonics technology. The goal is to lower power consumption but without compromising on the network throughput for various applications in photonic systems, especially concerning thermal issues. Thus, when advancing two variants of the algorithm, a kind of comparative research with the existing solutions in terms of developing the approaches is presented that are more efficient for minimizing the optical power loss and improving network performance. The presented approach brings focus to the need for efficient adaptive routing techniques [101], shown in Fig. 10, in the implementation of ONoc to reduce thermal effects and congestion, as well as further the development of on-chip communication architectures in future designs of chip multiprocessors.

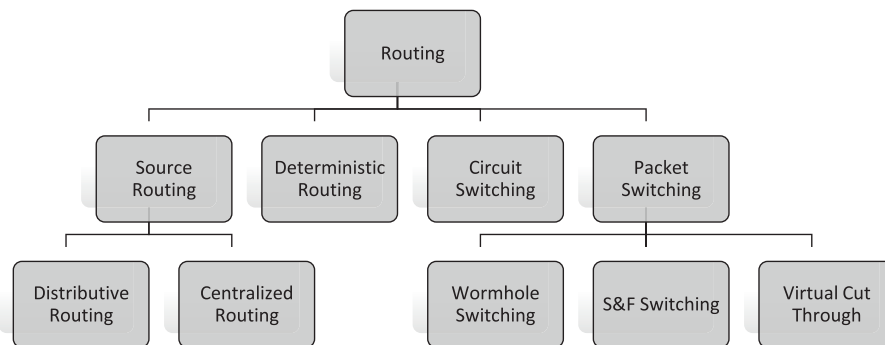


Figure 10: Hierarchical order of routing techniques

After improving the NoC mapping and scheduling region, the authors in [102] present a fault-tolerant task mapping method (FTTM) that can map and schedule both offline and runtime jobs on a 2-D mesh NoC and a modified NoC platform. The authors also provide a spare core technology in which the spare core may manage the communications related to the failed core in a 2-D mesh NoC and modified NoC (MNoC) platform in the event of a problem at a core. The suggested FTTM algorithm aims to save communication energy and maximize performance. A modified NoC platform is an excellent solution for this task because mapping and spare core placements are NP-complete. Tested against a variety of benchmarks, including synthetic and multimedia benchmarks, the presented FTTM algorithm on the improved NoC platform is contrasted with a 2-D mesh NoC and other comparable algorithms.

The fault-tolerant mapping algorithm (FTMAP) targeting NoC architectures with the purpose of increasing the system performance and energy efficiency after mapping the cores and replacing the faulty core with an available one, is presented in [103]. The steps of the algorithm include communication energy calculation, assignment of the vertices to processing elements, and communication energy update, as well as handling of busy and failed PEs. From the point of view of communication energy disposal, it is evident that FTMAP is an efficient protocol that aims to decrease the energy consumed so as to enhance the reliability and performance of a system. FTMAP provides an efficient solution for mapping cores that are tolerant to faults as well as having low communication energy, as compared to other mapping strategies. The work also focuses on the rationale behind FTMAP; however, machine learning approaches can be integrated to achieve potential improvements.

The work presented in [104] introduces a methodology to map the tasks at runtime within a bridge network-on-chip (BNoC) to achieve optimum energy consumption and deadlines. It provides mapping of networks and exhibits that BNoC topology offers steadier state results at an equivalent power-performance ratio as mesh networks. Extensive experimentations were conducted with the applications from the Parsec benchmark suite [105] and compared their suggested algorithm with all the basic mesh-based mapping algorithms. The intuitive location of the algorithm was found to be better than other existing techniques under a mesh-based environment with large power-performance differences. However, integration of thermal information into the mapping process is a weakness of this methodology.

4.2.3 Hybrid and Machine Learning-Based Mapping

This section provides an overview of Hybrid and Machine learning based mapping techniques. Hybrid mapping techniques in NoC architectures aim to combine the strengths of different algorithms or technologies to optimize performance metrics such as energy efficiency, fault tolerance, and communication latency. For instance, the GAPSO approach integrates Particle Swarm Optimization (PSO) with Genetic Algorithms (GA) to enhance communication efficiency using core graphs [45]. Similarly, other studies employ hybrid optimization algorithms based on double particle swarm in 3D NoC mapping [16], or utilize routing in 3D NoCs using Genetic Algorithms (GA) combined with PSO, optimizing communication paths and network efficiency [106].

Moreover, some studies enhance NoC-based MPSoC performance by adopting predictive approaches with Artificial Neural Networks (ANN) and guaranteed convergence arithmetic optimization algorithms, demonstrating significant improvements in performance metrics [107]. While some of these hybrid and ML-based methods are categorized under dynamic or static mappings based on their deployment model, they merit a separate discussion due to their methodological distinctiveness. Integrating these methods introduces trade-offs, such as improved adaptability vs. increased computational overhead or training time, which are discussed in the comparative analysis below.

4.2.4 Hybrid Mapping

Hybrid mapping combines elements of both dynamic and static mapping approaches to leverage their respective advantages. A hybrid exact and search-based mapping approach merges the precision of exact mapping with the scalability of search-based techniques, achieving a balance between accuracy and efficiency. Recent work, presented in [8] proposes algorithms that integrate both strategies for improved mapping outcomes in diverse NoC environments.

The process of application mapping on a NoC encompasses the allocation of application tasks to cores and the establishment of communication pathways between them. Optimal strategies for application mapping on a NoC need the utilization of machine learning algorithms to address the NP-hard mapping problem and prevent convergence to suboptimal solutions [22]. Strategies of hybrid task mapping that involve the team members in both the design stage mapping (before the task starts) and remapping during execution can be very efficient, thus reducing the communication and effort required for the task [8]. The Efficient Real-Time Embedded Application Mapping (ERTEAM) algorithm, which is intended to achieve minimum latency, fast simulation, and reduced communication energy, considers two factors: core average distance and communication energy efficiency [51]. Across a few methodical approaches, improvements in general metrics such as path tracing, communication estimation, power dissipation, and delay have proved to be the foundation of NoC-based multi-processor systems on chip (MPSoCs) [8].

Techniques for scenario-based hybrid application mapping (HAM) are the main focus of the work described in [108]. The application input space is deliberately grouped such that, when processed under the same operational points, data inside the same scenario show similarities regarding workload. It has been shown that the static grouping of the input space into data scenarios is a useful abstraction layer for simplifying the creation and use of better run-time managers. But current hybrid application mapping methods ignore the mutually reinforcing effect of mapping choice and the use of dynamic voltage/frequency scaling (DVFS) for workload adaptation. Variations in the input can be compensated for either by fully re-mapping the application, which could result in a high reconfiguration overhead, or by simply altering the resource's DVFS settings, which provides a low-overhead adaptation option and significantly reduces the required overhead when compared to DVFS-agnostic HAM. Additionally, by employing low-frequency DVFS settings, DVFS allows a mapped application to be finely tuned to the variations in input data. It is demonstrated that, particularly in the presence of data situations, this combination strategy may save even more energy than a single mapping adaptation scheme. Furthermore, as they show in an empirical study based on four different applications and three different architectures, they propose two additional scenario-based DVFS-aware HAM approaches that consistently outperform existing mapping approaches in terms of the number of deadline misses and energy consumption. Furthermore, it is demonstrated that these advantages hold for target designs with rising mapping migration overheads, restricting the need for frequent reconfigurations of the mapping.

4.2.5 Machine Learning Based Mapping

Machine learning, along with deep learning and artificial intelligence, has revolutionized the common person's needs and demands in almost every area. This trend is also present in application mapping in NoC. The following is the latest literature related to machine learning in application mapping.

In [75], the combination and application of deep reinforcement learning to NoC for auto-configurable NoCs. The gradual and dynamic volte-face of values, levels, and frequencies brings to light a glimpse of machine learning potential, which can be used for the resolution of power and performance problems in NoC-based computing systems.

The FANC algorithm, a machine learning method, plays a great role in lowering communication costs and improving several results of the simulation [76]. The latter renders the search dynamics much simpler, which provides a favorable pathway for the architecture of NoC-based systems that suffer from high integration density-related issues.

The concept of implementing the Machine Learning (ML) methods into the NoC architecture has demonstrated a high promise of increasing the efficiency of routing, latency, and energy performance in the next-generation SoC-NoC systems [109]. In the meantime, recent research shows that, although ML can be quite useful to optimize particular NoC elements, including arbitration in a high-contention environment, much human effort is still necessary in the global architectural development [110].

The latest article presents and demonstrates RL-MAP, a new approach that uses RL for NoC application mapping. RL-MAP applies a combination of Actor and Critical neural networks, which leads to a better result on the Xilinx Zynq UltraScale + MPSoC ZCU 104 evaluation kit [20]. This method takes advantage of the weaknesses of traditional strategy and stresses the importance of machine learning to replace the positioning systems.

In [75], the combination and application of deep reinforcement learning to NoC for auto-configurable NoCs. The gradual and dynamic volte-face of values, levels, and frequencies brings to light a glimpse of machine learning potential, which can be used for the resolution of power and performance problems in NoC-based computing systems.

The FANC algorithm, a machine learning method, plays a great role in lowering communication costs and improving several results of the simulation [76]. The latter renders the search dynamics much simpler, which provides a favorable pathway for the architecture of NoC-based systems that suffer from high integration density-related issues.

To shed light on the fact that the mapping in NoC is NP hard, it proposes a machine learning approach resulting in the state-of-the-art mapping accuracy score [22]. Although the approach to machine learning methods results in performance degradation from incorrect predictions, this act necessitates the replacement of heuristics by machine learning for more efficient NoC design.

The task of mapping deep learning applications across NoC is described, with the introduction of a multilevel approach being analyzed using OCTAVE simulations, which have been proven to be effective [13]. The outcome proves the advancement in throughput, latency, and energy consumption, hence the prospects that this can possibly open up for further performance enhancement of NoC for real-time applications, which the demands of AI technologies, etc., also affect. Table 2 shows the application mapping categories.

Table 2: Application mapping categories

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
	Reinforcement Learning (RL)	Improves NoC performance	Previous methods have limitations	RL-MAP method using Actor-Critic networks and local search algorithm	Xilinx Zynq UltraScale+ MPSoC ZCU104 Evaluation Kit	Latency and throughput improvement	[20]
	Deep reinforcement learning	Improves energy efficiency and performance	Configuring NoC resources efficiently	RL agents dynamically adjust voltage/frequency levels	NoCs of various sizes	30%–40% improvement in energy-delay product	[75]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
	FANC (Fault-Tolerant Multi-Application Mapping)	Significant reduction in communication costs	Simplifies the search process in design exploration	ML-based model for enhanced search process	Various simulation parameters	266% reduction in communication costs	[76]
	Multilevel mapping approach	Improves throughput, latency, and energy consumption	Focuses on mapping deep learning applications	OCTAVE simulations	NoC architectures	Outperforms direct mapping techniques	[13]
	Machine learning model	Achieves high mapping accuracy	Incorrect predictions may lead to performance loss	Low-complexity non-symmetric NoC mapping dataset	–	At least 99.6% accuracy, 96.3% average mapping accuracy	[22]
	DREAM NoC	Reduces latency, increases throughput	Limited evaluation scenarios	Ant colony optimisation + RL for routing	Irregular topologies under uniform random traffic	63% reduction in latency, 25% increase in throughput	[111]
	IWOA-IGA	Energy and latency reduction	A combination of algorithms requires more computational resources	Hybrid optimisation (WOA) and (GA)	Approach minimises energy consumption, latency, communication costs, and power usage	Significant improvements in communication cost and overall performance metrics	[112]
	GAMS	Minimised communication overhead, Adaptability	complexity in design and execution	(GA) for optimising application mapping in a 3D hybrid NoC	Row-stationary dataflow approach to create a data movement energy consumption model	Enhancing acceleration performance in deep convolutional neural networks	[113]
	Mixed-Integer Linear Programming (MILP)	Energy reduction, adaptability	Require careful tuning to achieve optimal performance	Task-Resource Co-Allocation in mapping Deep Neural Networks (DNNs)	Addresses the unique communication patterns of various DNNs by formulating an MILP-based mapping algorithm	Performance of DNNs on NoCs, confirming their effectiveness in enabling power-efficient execution	[114]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
Machine Learning-Based Mapping	HAS-RL	Optimisation of resource costs	Accurate characterisations of quality loss and data rates	RL for optimising Neural Network (NN) accelerators.	Establishes the relationship between quality loss and data approximation rate	Improve the performance by managing the trade-off between result quality and communication delay	[115]
	NCTPAM	Congestion management, optimisation of 3D NoC	Dependence on learning accuracy	Reinforcement Learning for optimizing application mapping in 3D NoC	Application mapping to determine task allocation	To optimise application mapping and resource allocation	[116]
	CSAP	Cost-effectiveness	Complexity of traffic management	Arbitration policy designed to reduce latency in communication for DNN	Monitors and adjusts the rate of packets sent from each source node to minimise execution time.	Improves performance by addressing synchronisation issues and optimising traffic patterns	[117]
	HP-LSM	Effectively control congestion in hotspot regions	Additional hardware components might increase the overall resource requirements	Hotspot prediction using LSM integrated to enhance performance.	Spiking neural network (SNN) for monitoring and predicting the hotspot	High accuracy in predicting hotspot generation	[118]
	Hyb ACOA-3D-NoC (Artificial Humming Bird + Coati Optimization)	Significant power, delay, and execution-time reduction; suitable for 3D NoC	Metaheuristic tuning required; design-time only	Hybrid bio-inspired optimization	Power-aware application mapping using hybrid AHB + Coati optimization	Up to 29% lower power, 35% lower delay vs. state-of-the-art	[80]
	Genetic algorithm and ant colony optimisation	Lower communication expenses	Potentially increase computational demands.	Combining GA and ACO to optimise the mapping	Automating the mapping process to enhance system stability and reduce design overhead	Improves communication efficiency in NoC architectures	[119]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
	Reinforcement-Guided Genetic Algorithm (GA+RL with XAI)	Significant reduction in communication cost; faster convergence; explainable decisions	Higher design-time complexity; not runtime adaptive	Hybrid GA optimized via Reinforcement Learning with SHAP-based XAI	RL agent embedded in GA loop to guide selection, crossover, and mutation using mapping-quality feedback	>52% reduction in communication cost; lower variance; improved convergence stability	[69]
	SaHNoC algorithm	Optimal energy efficiency	Wireless switches consume more power	Self-Adaptive Hybrid NoC structure	2D mesh architecture	Latency and throughput improvement	[72]
	SDR (Smart Dependable Routing)	Strong security; fault tolerance; deadlock-free routing; scalable	Routing-only (no task mapping)	Secure adaptive routing	Security zones with fault-aware path selection in 2D mesh NoCs	Improved security, dependability, and scalability under faults	[120]
	ERT-EAM strategy	Improved throughput and latency	Focuses on real-time embedded applications	Minimum Core Average Distance (CAD) and Xilinx Vivado 2020.2 software	Xilinx Zynq UltraScale+ MPSoC ZCU104 Evaluation Kit	Better delay and area metrics	[51]
	Multi-Objective Particle Swarm Optimisation (MOPSO) + RL	Improves reliability, reduces energy consumption	Aimed at addressing permanent PE failures	Intelligent agents trained through RL	Heterogeneous NoCs	Enhanced reliability, reduced energy consumption	[32]
	FT-PDC	High fault tolerance; congestion awareness; shortest-path selection	Increased routing logic complexity	Fault- & congestion-aware adaptive routing	Path diversity with traffic-aware routing in 3D mesh NoCs	Lower latency and higher throughput than recent methods	[121]
	Quadratic programming QP-based approach	Router placement affects communication cost	High re-transmissions	SA	NoCs of various sizes	Throughput improvement	[66]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
Dynamic Mapping	RLARA	TSV-aware; adaptive; improved reliability and throughput	RL overhead; routing table management	Reinforcement learning-assisted routing	TSV-aware Q-learning with congestion prediction and K-means compression	Up to 9.04% lower delay; 8.58% higher delivery rate	[122]
	RL-FTR	Handles link/router faults; FPGA validated; adaptive	Training and hardware overhead	Reinforcement learning-based routing	RL-driven path selection for fault-tolerant mesh NoC routing	Optimal routing paths under multiple fault scenarios	[123]
	Q-Function-Based Adaptive Routing	Balanced inter-layer traffic; thermal mitigation	Learning convergence time	Q-learning-based routing	Q-function with RTM and topology awareness	14.0%–28.2% throughput improvement; better thermal balance	[124]
	Ranking and selection (RSM)	Fault-tolerant application mapping	The technique has a lot of computational delay	Dropping Neurons Depending on their degrees of contribution	Eliminating a few neurons	Latency and throughput improvement	[40]
	Ring NoC design	Strong performance, efficient data transfer	Limited to ring topology	Design concept, simulation in Xilinx ISE 14.7	Virtex-5 FPGA hardware	Latency and throughput improvement	[77]
	2D and 3D Mesh NoC	Scalable and architecturally flexible	Limited to mesh topology	Design and simulation using Xilinx ISE 14.7 and Modelsim 10.0	Virtex-5 FPGA hardware	Latency and throughput improvement	[78]
	GAPSO (Particle swarm optimization + Genetic Algorithm)	Optimises communication efficiency	NP-hard problem of mapping operation	Experiments on Cores Graphs	Performs better than PSO alone	Latency and throughput improvement	[89]
	Double particle swarm optimisation	Optimises heat dissipation, lowers latency	Risk of falling into local optimal solutions	Region-based IP core to NoC mapping	$4 \times 4 \times 4$ 3D NoC scale	Decreased energy loss, better throughput	[16]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
Static Mapping	Adaptive control message distribution	Increases bandwidth utilisation	Increased energy usage compared to a single NoC	Dynamic switching between networks based on runtime consumption	Multiple NoCs	Up to 72.7% more energy usage	[90]
	Heuristics	Improve processor core ageing and communication overhead	Lacks in reducing power consumption	Finding areas with less congestion and heating issues	The approach operates by avoiding hotspot places	Throughput improvement	[94]
	ABC	Reduced energy	Computational delay is high	Reduce overall energy usage	Reduce energy use	Better than CastNet	[95]
	Lightweight application mapping and scheduling	Energy efficiency, dynamic adjustment	Overhead	Application mapping and scheduling in multicore hybrid wireless	Core usage thresholds and leverages long-distance wireless communication	Energy consumption, latency, deadline violations	[96]
	Linear programming	Energy-aware mapping	Performance may vary in various applications	Linear Programming Combined with Simulated Annealing	SA and GA are integrated with the linear programming model to optimize mapping under energy constraints	Performed better in a 2D mesh	[97]
	Mixed-Integer Linear Programming (MILP)	Energy efficiency, reliability	Heuristic limitations	3-Stage Heuristic Approach for Task Mapping	(DVFS) to minimise energy consumption	Reduced contention and energy savings	[99]
	Branch and Bound algorithm	Thermal management, power efficiency	Adaptive routing challenges, computationally intensive	Branch and Bound Algorithm Optical NoC	Addressing thermal issues in the photonic system	Minimising power loss, enhancing performance	[100]
	FTTM	Fault tolerance, energy efficiency	FTTM algorithm and managing spare cores add complexity	Fault-tolerant task mapping and scheduling for 2-D mesh and modified NoC	Schedules both offline and runtime jobs while incorporating spare core technology	Improved performance on the modified NoC compared to the 2-D mesh	[102]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
Hybrid Mapping	FTMAP	Energy efficiency, enhanced performance	Machine learning could enhance its capabilities further	Aimed at improving system performance and energy efficiency.	Communication Energy Calculation, Handle faults	Low communication energy	[103]
	BNoC	Energy optimisation and efficient	Thermal management	Runtime task mapping to optimize energy consumption and meet deadlines	More stable state results at a comparable power-performance ratio	Improvements in power-performance	[104]
	CTH topology	Lowers mapping overhead, reduces execution time	Limited to specific topologies	Runtime Application Mapping Based on Cube-Tree-Hybrid (CTH)	Various networks, including mesh and torus.	15%–21% gains in overall energy usage	[39]
	Genetic Algorithm-based Mapping Strategy (GAMS)	Improves PE utilisation; enhances mapping flexibility; optimises energy consumption; increases execution speed	Requires complex configuration and computation overhead	Genetic algorithm	Novel mapping strategy integrated with 3D Hybrid Optical-Electrical NoC (3DHOENoC)	15.9% improvement in execution speed; better communication and energy efficiency	[113]
	Scenario-Based DVFS-Aware Hybrid Application Mapping (HAM)	Reduces energy consumption and deadline misses; adapts dynamically to workload variations	Higher overhead for frequent re-mapping in large architectures	DVFS-aware mapping	Combines design-time optimisation with runtime adaptation using scenario-based clustering	Significant reduction in energy consumption and improved deadline adherence	[108]
Hybrid Mapping	Double Particle Swarm Optimisation (DPSO)	Higher throughput rate, lower energy loss; avoids local minima	Relatively complex hybridisation of algorithms	Particle swarm + Genetic algorithm hybrid	3D NoC partitioning and mapping via DPSO-based heuristic algorithm	Better throughput and reduced energy loss compared to traditional heuristics	[16]

(Continued)

Table 2 (continued)

Category	Algorithm	Pros	Cons	Technique	Method	Outcome	Ref
	HyDra framework (Hybrid Task Mapping)	Reduces communication and energy cost; adapts to dynamic workloads; lower latency	Requires pre-generation of multiple design-time mappings	Design-time + Runtime mapping	Combines static design-time mapping with dynamic runtime remapping	14% reduction in communication cost; 15% less energy; 19% latency reduction	[8]
	Improved Whale Optimisation with Genetic Algorithm (IWOA-IGA)	Enhances convergence speed; reduces power, energy, and latency; better for large task graphs	Algorithm complexity requires parameter tuning	Whale optimisation + Genetic algorithm hybrid	Integrates improved whale optimisation with a refined genetic algorithm	Significant improvement in energy, power, and latency; superior convergence	[112]

The study suggested a packet Routing way called DREAM NoC and Distributed Reinforcement Learning [111] was used for notifying in switched systems containing hard real-time. Can be attributed to the decline latency and the rises throughput indicate that adaptive NoCs could be particularly prudent in changing circumstances.

Faster mapping onto 2D NoCs and performance metrics are effectively traded off using the whale optimization algorithm with improved genetic mechanism, IWOA-IGA [112]. The presented approach, a hybrid mechanism, minimizes energy, latency, communication costs, and power while optimizing mapping results. Its foundation is an enhanced whale framework with GA features, and an appropriate approach for NoC application mapping is provided. A ranked selection-based approach is used to offer improved initial mapping for IWO, direction-based crossover, and mutation ability throughout GA evolution. The final mapping method performs better overall in terms of communication cost.

In order to optimize energy usage, an application mapping technique based on the genetic algorithm (GAMS) is presented in [113], Genetic algorithm (GA) is used to find the optimal mapping solution utilizing row-stationary dataflow by building a data movement energy consumption model during processing and integrating various network topology characteristics and convolution parameters. The assessment and simulation findings demonstrate that GAMS can offer a more adaptable mapping technique that leads to higher acceleration performance by maximizing the usage of processing units while lowering processing energy consumption. To further enhance deep convolutional neural networks' processing speed and communication efficiency, a 3D hybrid optical-electrical NoC is presented. GAMS allows for the flexible mapping of the convolution layer into the processing elements array, without being constrained by the topology, size, or parameters of the processing elements array. GAMS can lower the overhead energy consumption from data movement and improve the reusability of data. Acceleration performance may be enhanced by fully using processing elements. GA may be used to find the model's optimal solution and provide GAMS mapping results. The energy consumption cost of alternative mapping techniques may also be assessed using the energy consumption model. A 3DHOENoC has a minimal communication architecture, excellent

adaptability, and unidirectional communication. The energy model is presented in order to enhance the GAMS processing convolution layer acceleration performance.

The work presented in [114] highlights the implementation of NoCs as a means to enhance the communication of deep neural networks (DNNs) and, at the same time, reduce energy consumption. This emphasizes the issue of typical mapping processes to effectively execute DNNs on NoCs since various DNNs exhibit distinct communication patterns. The authors present a mixed integer linear programming (MILP)-based task-resource co-allocation formulation and employ it in mapping of NNs to NoC-based multicore systems. It proposed a mapping algorithm enhanced with their neighbor-aware and adapted simulated annealing and genetic algorithms for an efficient mapping of DNNs onto NoCs. These results confirm the appreciability of the proposed approaches in enabling the platform to speed up DNNs across NoCs power-efficient manner.

In another study, restrictions and opportunities of mapping applications onto NoC 3D are proposed in [116]. The authors presented the NeurMap3D neural mapping model as an application of reinforcement learning for the creation of ICs based on the 3D NoC architecture optimization [86]. Also, the proposed work introduces the neural congestion-aware through-Silicon vias placement and application mapping (NCTPAM) approach, which consists of application mapping, vertical link placement, and load balancing over specified vertical links for applications. The presented techniques are intended for improving the NCTPAM algorithm to minimize the CPU time in the framework of the 3D NoC.

A communication synchronization-aware arbitration policy (CSAP) has been implemented in NoC-based deep neural network accelerators and aims to reduce latency by attempting to synchronize the communication between the processing elements of the connective layers [117]. CSAP utilizes a negative feedback mechanism on the rate of the packets sent by each source node to trim down the execution time over the above-mentioned neural networks of various sizes in comparison with the given local-age-based policy. The presented policy costs only 1% of the total policy charges, which is 3% of the total charges in the state. To achieve this, an additional hardware cost of 6% of the existing router is required to cater for incremental overhead hardware. Based on the findings of this approach explores traffic patterns in neural network accelerators and synchronization issues to improve its performance.

The authors of [118] presented an HP-LSM, also known as hotspot prediction, employing a liquid state machine based on NoC routers. Utilizing a liquid state machine (LSM) based predictor to train it extremely accurately, a spiking neural network (SNN) is used in the observe method of the HP-LSM software model. The LSM uses the multi-layer perceptron (MLP) to forecast router density in the next time step, starting with states based on spiking. Two hardware components of HP-LSM, the LSM and MLP, are designed for dynamic evaluation of possible NoC hotspot generation. Furthermore, a potent routing algorithm called Heed Maps (HaPM), is included to enhance NoC performance by skillfully controlling congestion in hot spot areas.

The work presented in [119] explores the issue of effectively partitioning application software components to organize onto the NoC architecture in an effort to reduce communication expenses. It points out the issues that arise when dealing with SoC applications, which makes this process difficult. To reduce the overhead required for the NoC design, the present work develops a new approach that combines Genetic Algorithm and Ant Colony Optimization for the mapping process. Thus, it is challenging to increase the efficiency of using these two algorithms for obtaining the best matching solutions. The study also underlines the crucial importance of automation when it comes to the mapping process, which will have an impact on the stability of the system.

Dynamic, static, and hybrid mapping strategies, to the more advanced machine learning strategies, all the approaches solve major challenges such as congestion, fault tolerance, and thermal management. By

the combination of reinforcement learning, genetic algorithms, and other methods, it has been established that NoC-based systems have been enhanced in efficiency, particularly when used in complicated and diverse networks. The configuration of these strategies for future research must persist, as they still have improvements within their considerations in trying to justify the existing limitations and the emerging requirements of multi-processor systems-on-chip (MPSoCs) and neuromorphic computing architectures. Thus, it is possible to establish that using the optimized aspects of both conventional and innovative approaches enables the development of more effective, reliable, and extensive NoCs.

4.3 Comparative Insights and Trends in Mapping Techniques

While the previous sections detailed various mapping techniques, this section highlights observable trends and performance patterns across the literature. A review of the selected 75 studies indicates that machine learning-based techniques have seen a significant rise in adoption in recent years, accounting for approximately 28% of all reviewed methods, followed by heuristic/metaheuristic techniques (24%), and hybrid mapping strategies (17%). Static mapping techniques remain relevant for systems with fixed workloads but are gradually being supplemented by adaptive approaches.

In terms of performance metrics, energy efficiency and latency remain the most frequently evaluated criteria, appearing in over 65% of the studies. In comparison, reliability and thermal management were reported in approximately 30% and 22% of cases, respectively. Notably, dynamic and ML-based approaches generally outperform others in handling fault tolerance and real-time reconfigurations, but at the cost of increased runtime complexity. [Table 3](#) provides a breakdown of technique usage and associated performance focus across the reviewed literature.

Table 3: Frequency and performance focus of mapping approaches in NoC literature

Technique type	% of papers	Common performance focus	Typical trade-offs
Machine learning	28	Energy, latency, fault tolerance	High complexity, training overhead
Heuristic/Metaheuristic	24	Energy, throughput	Near-optimal but no guarantees
Hybrid mapping	17	Adaptability, energy	Integration complexity
Static mapping	21	Predictability, low overhead	Inflexibility to runtime changes
Exact algorithms	10	Optimality	Scalability issues, high computation

Conclusion: As application mapping techniques in NoC architectures, there are diverse strategies with particular advantages and drawbacks. Dynamic mapping techniques offer high flexibility and are fault-tolerant to different dynamic conditions at runtime, but come with a cost of runtime overhead. They have predictable performance and low cost at runtime, but are non-flexible and are most suitable for systems with known workloads since they provide static mapping approaches. Hybrid approaches are designed to reach maximum efficiency with minimum use of an individual method; however they come at the price of higher complexity. On the other hand, machine learning-based methods integrate intelligence and the capability to learn, add time to adapt and requires more resources for computing. A comparative summary of these strategies is provided in [Table 1](#).

Just to sum it all up, [Table 2](#) lists the advantages and disadvantages of each mapping method. Machine learning methods, for example, might be a promising approach, but they need a lot of training data and computing power, and both might not be available. Besides, their decision-making process is not so clear, which makes it hard to trust in critical systems. Static methods are easier to maintain and predictable, but dirty for runtime changes and unpredictable faults. Although a few articles showed that hybrid methods outperformed approaches that exclusively relied on static or dynamic mapping by integrating both, other studies pointed out that hybrid methods added overheads and did not have advantages over pure static or dynamic methods under the specific workstations. This illuminates that there is no one-size-fits-all approach. A lot of work needs to be done to scale these algorithms, to lower their energy footprint, and to make the mapping from quantum to classical architectures more robust, particularly as systems grow in complexity.

Even though many techniques have been proposed, some areas still need more attention from researchers. One major issue is scalability: most current algorithms don't perform well when the number of cores exceeds a certain threshold. In *Quadrisection-Based Task Mapping on Many-Core Processors to Energy-Efficient On-Chip Communication* [125], it is stated that, with an increase in the number of cores (many-core), mapping and communication costs increase. Another challenge is energy efficiency, because many mapping strategies focus solely on performance and ignore the power the system consumes. Also, not enough work has been done on security, especially in cases where data can be attacked through mapping vulnerabilities or side channels. Most existing solutions also assume ideal conditions, but in real-world systems, faults, delays, and temperature variations can affect the mapping. There is a strong need for more practical, fault-aware mapping methods that perform reliably in actual chip environments.

5 Application Mapping Performance Metrics

Many application-mapping techniques have been proposed in recent years. Most of the applications are extremely promising. To evaluate mapping algorithms, various performance metrics are used, including latency, throughput, packet drop rate, bandwidth utilisation, response time, and error rates [126].

In NoC architectures, processing elements communicate to complete tasks distributed across the chip. Latency directly affects the time required for communication exchanges between PEs. When mapping applications onto the NoC, it is important to consider the latency presented by the routing paths between processing elements. Applications requiring fast task completion or real-time responsiveness must be strategically mapped to minimise latency in communication pathways [127]. The formula to calculate latency is given in [Eq. \(1\)](#).

$$L = (\text{Response Time} - \text{Transmission Time}) \quad (1)$$

In NoC architectures, throughput [128] is the rate at which data is transferred between processing elements within the chip. It is crucial, when mapping an application, to give optimal performance in sending data between PEs [129]. The method to calculate throughput is given in [Eq. \(2\)](#).

$$T = \text{Data Transferred} / \text{Time} \quad (2)$$

In NoC, data can be corrupted or incomplete due to packet loss during transmission among processing elements. This is crucial to minimise packet loss while mapping the application to the NoC. Optimising the routing path, adjusting the buffer size, and implementing error detection and correction strategies to reduce packet loss [130]. The technique to calculate packet loss is given in [Eq. \(3\)](#).

$$PL = (\text{Number of Lost Packets} / \text{Total Number of Packets}) * 100 \quad (3)$$

Efficient use of resources, such as communication channels and interconnects, is considered bandwidth utilisation when implementing application mapping [128,131]. It should be considered when mapping an application onto the NoC that there must be an effective and balanced use of resources to meet the communication demands of the tasks [132]. Eq. (4) represents the formula for the calculation of bandwidth utilisation.

$$BU = (Data\ Transferred / Maximum\ Bandwidth) * 100 \quad (4)$$

In NoC architectures, the response time is influenced by several factors such as network topology, routing algorithms, network congestion, and flow control. The motive is to minimise response time, and resources are allocated with this in mind [133]. The method to calculate response time is given in Eq. (5).

$$RT = Processing\ Time + Transmission\ Time + Queuing\ Time + Wait\ Time \quad (5)$$

Error rate in NoC architectures refers to the frequency of errors encountered during communication between processing elements. Strategies such as redundant mapping, where critical tasks are duplicated across multiple PEs, can enhance fault tolerance and reduce the overall error rate by providing backup resources to handle errors or failures [134]. The formula to calculate the error rate is given in Eq. (6).

$$ER = (Number\ of\ Errors / Total\ Number\ of\ Requests) * 100 \quad (6)$$

6 Discussion and Future Work

Application Mapping and Scheduling have emerged as the key levers to drive performance, scalability, and reliability in 3D Network-on-Chip (NoC) systems, which are a target for massively parallel megacore designs. In the literature surveyed, the solution space remains broadly divided into property families—static, dynamic, hybrid, and AI-driven (learning-based) all of which are characterized by their unique advantages and disadvantages when accounting for 3D-specific constraints, such as TSV/ILV structures, vertical traffic concentration, thermal gradients between layers, and fault clustering in dense stacks [27,29,135]

Static mapping is still desirable for its predictability and low runtime overhead, but it is increasingly inadequate in 3D environments, where placements may be optimized based on evolving heat and congestion that directly depend on placement decisions. Recent work employing RL-driven task migration for runtime adaptation indicates an ability to reduce peak temperature and stabilize execution, benefits that static maps fail to sustain in the face of real workload variability [27,29]. On the other hand, dynamic mapping and online migration increase robustness through adaptation to runtime telemetry data (e.g., queueing delay, router load, thermal sensors), while still being lightweight to avoid control-plane overhead that can obviate the performance benefits they seek to deliver [29,133].

One near-term solution is hybrid mapping: compute a strong offline back-bone communication with thermal constraints, and then provide online constrained corrections (migration, local remapping) only where required by telemetry signals hotspots, congestion bursts, or faults [27,29]. This hybrid logic also matches trends in chiplet-based fabrics, where non-uniformity (inter-chiplet latencies, cache or memory locality) makes pure global centralized runtime decisions expensive. We show how recent chiplet runtime systems [133,135] provide chiplet-aware scheduling or migration policies that can considerably increase throughput with respect to memory-intensive workloads, corroborating the need for hierarchical mapping (cluster → chiplet → PE) in future 3D NoC and chiplet-interconnect platforms [136–138].

On the AI side, learning-based mapping saturates toward modeling high-dimensional state representation and long-range dependencies. As a specific example, transformer-based reinforcement learning

for fault-tolerant application mapping focuses on deriving robust mappings under failure scenarios while constraining communication cost and latency [1,30]. Open questions persist, though, such as training cost, data efficiency, safety/constraint satisfaction at deployment, and reproducible evaluation protocols that enable fair comparisons against classical heuristics [1,30].

Fault tolerance and security are no longer just an optional add-on. In large systems, faults cannot be avoided, and recent NoC research regularly presents new fault models and diagnosis mechanisms to keep the topology and thereby low transmission failure under clustered faults [139]. At the same time, security-oblivious decision frameworks explicit in task-mapping methods, under security, adaptivity, and overhead criteria for 3D-NoC MPSoCs, are evaluated in an IoT environment [137], whilst this excessive DoS-like attack strain on NoC routers is referred to by router-level prioritization frameworks [140,141]. These threads suggest that forecasting mapping frameworks must consider fault/security objectives as first-class, instead of a post-processing action [139–141].

Lastly, the evaluation stack itself is turning into a research problem. Recent efforts to trace realistic applications and provide physical fidelity for higher-fidelity network simulation abstract many details, making task-graph modeling more relevant for consideration; these works have also called for coherent and realistic traffic generators to subject mapping claims to relevant workloads [142,143]. In the absence, the improvements from ML-based mapping results become confined to tight benchmark and simulator settings without translating to end-to-end improvements [142,143].

High-Priority Future Directions

Nonetheless, there exists a real-time AI-based NoC controller that takes a constraint-aware, lightweight AI approach (latency/throughput + thermal caps + fault/security constraints) [29,30,140].

Examples include hierarchical hybrid mapping for 3D stacks and chiplets (offline backbone + online corrections) to go beyond centralized control [136–138].

First, it enables standardized, reproducible benchmarking with realistic task graphs and traffic models that compare ML vs. classical methods fairly [1,142,143].

Embedded fault-tolerance: fault models, diagnosis or repair-aware mapping, and security-aware routing/mapping in the main decision loop [135,139–141].

7 Limitations of Current Techniques

Although reinforcement learning has been demonstrated to be effective in mapping applications, it also has its issues that limit practical application. A big challenge is the training overhead RL requires a large number of iterations and simulation episodes to converge, which may take a long time and be expensive in terms of computation, especially for the large NoC system. If the environment is dynamic, then it gets even more complicated as you might have to re-train if a shift in workload or network condition occurs. Another issue is real-time adaptability. While RL is capable of learning (and un-learning) to dynamically adapt to changes in its environment, the inference time of an RL policy must be extremely low if we want to deploy it in a live system, which is not always the case. When considering fault tolerance, the vast majority of RL-based methods assume a fixed topology that is stationary. In case of a link/core failure, the agent will not be able to recover quickly/at all unless it was trained to do so explicitly. This brings into question the robustness of RL in environments where faults occur or where the behavior of units is difficult to predict. In conclusion, whereas the flexibility of RL is proven, its scalability, latency, and robustness are still major open questions of the field in need of further investigation and optimization.

8 Conclusion

Finally, this study provides a structured review of application-mapping approaches and techniques that were introduced in recent years, specifically for 2D and 3D NoC building blocks or from a general mapping perspective that can be used for any suitable network topology. We classified the techniques by algorithm-based functionality, such as static, dynamic, machine learning-based, power-aware, fault-tolerant, and hybrid techniques. These approaches have unique energy efficiency, load balancing, scalability, and flexibility trade-offs.

Conclusions show that the static approach provides simplicity and predictability, but at the price of flexibility. On the one hand, methods that are dynamic or based on machine learning support adjusting in real-time, but usually at the expense of complexity or computational cost. An exciting avenue for future work is hybrid approaches, which combine aspects of multiple methods for a wider range of usable scenarios.

This review is limited in spite of the extensive coverage. No quantitative meta-analysis or statistical comparison of mapping strategies was included, and recent developments focusing on industry have likely been missed. Performance metrics used across studies were not standardised, which also made direct comparison difficult.

Scalable hybrid approaches, security-aware mapping, and lightweight learning architectures for edge environments remain as some areas for future research. In addition, there is a requirement for benchmark datasets and evaluation frameworks to align the evaluation criteria across different studies.

This survey serves as a starting point for researchers to understand important trends, challenges, and gaps in NoC application mapping. Next-generation mapping solutions can gain from intelligence and fault-tolerant strategies, which can combine to address the complex mapping requirements of high-performance NoC systems in a better way.

When analysing the effectiveness of application mapping techniques in NoC architectures, one needs to understand a set of key parameters, such as latency, throughput, packet loss, bandwidth consumption, response time, and error rate. These metrics help enhance the effectiveness and reliability of NoC systems by properly mapping applications to meet the various requirements of application mapping. The researcher can design and implement better solutions by using performance metrics, thereby increasing the functionality and reliability of NoC architectures.

Acknowledgement: The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number “NBU-FFR-2025-2903-09”.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Naveed Ahmad led the literature search and screening, designed the review methodology, compiled the figures and tables, synthesised the findings, and drafted the original manuscript. Muhammad Kaleem contributed to literature verification, supported the screening process, refined the taxonomy, and participated in reviewing and editing the manuscript. Mourad Elloumi provided technical oversight of the mapping classifications, validated extracted data, and contributed to the structural and analytical revision of the paper. Muhammad Azhar Mushtaq assisted with reference verification, expanding related work, and proofreading to ensure consistency and clarity. Ahlem Fatnassi contributed to the thematic grouping of the reviewed studies, improved the tables and figures, and participated in manuscript editing. Mohd Fazil contributed to the analysis of performance metrics and assisted in harmonising the content across sections. Anas Bilal supervised the overall review process, provided methodological guidance, validated the revisions, contributed to the interpretation of the results, and handled project administration.

Abdulbasit A. Darem supported the classification of algorithms, assisted with proofreading, and contributed to consistency checking and manuscript editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: This article is a review paper and, as such, does not contain any new data. All information and data cited in this manuscript are derived from previously published research articles, which are referenced accordingly throughout the text. The availability of the data and materials is documented in the respective cited sources.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Kaur SP, Ghose M, Pathak A, Patole R. A survey on mapping and scheduling techniques for 3D Network-on-Chip. *J Syst Archit.* 2024;147:103064. doi:10.1016/j.sysarc.2024.103064.
2. Sahu PK, Chattopadhyay S. A survey on application mapping strategies for Network-on-Chip design. *J Syst Archit.* 2013;59(1):60–76. doi:10.1016/j.sysarc.2012.10.004.
3. Murali S, De Micheli G. Bandwidth-constrained mapping of cores onto NoC architectures. In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*; 2004 Feb 16–20; Paris, France. p. 896–901.
4. Hu J, Marculescu R. Energy-aware mapping for tile-based NoC architectures under performance constraints. In: *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*; 2003 Jan 21–24; Kitakyushu, Japan. p. 233–9.
5. Murali S, Benini L, De Micheli G. Mapping and physical planning of networks-on-chip architectures with quality-of-service guarantees. In: *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*; 2005 Jan 18–21; Shanghai, China. p. 27–32.
6. Hu J, Marculescu R. Energy- and performance-aware mapping for regular NoC architectures. *IEEE Trans Comput-Aided Des Integr Circuits Syst.* 2005;24(4):551–62. doi:10.1109/tcad.2005.844106.
7. Chou CL, Marculescu R. Contention-aware application mapping for Network-on-Chip communication architectures. In: *Proceedings of the 2008 IEEE International Conference on Computer Design*; 2008 Oct 12–15; Lake Tahoe, CA, USA. p. 164–9. doi:10.1109/iccd.2008.4751856.
8. Amin W, Hussain F, Anjum S, Saleem S, Ahmad W, Hussain M. HyDra: hybrid task mapping application framework for NOC-based MPSoCs. *IEEE Access.* 2023;11:52309–26. doi:10.1109/access.2023.3279501.
9. Ababei C, Katti R. Achieving network on chip fault tolerance by adaptive remapping. In: *Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing*; 2009 May 23–29; Rome, Italy. p. 1–4. doi:10.1109/ipdps.2009.5161202.
10. Derin O, Kabakci D, Fiorin L. Online task remapping strategies for fault-tolerant Network-on-Chip multiprocessors. In: *Proceedings of the Fifth ACM/IEEE International Symposium on Networks-on-Chip*; 2011 May 1–4; Pittsburgh, PA, USA. p. 129–36. doi:10.1145/1999946.1999967.
11. Chang YC, Gong CA, Chiu CT. Fault-tolerant mesh-based NoC with router-level redundancy. *J Signal Process Syst.* 2020;92(4):345–55. doi:10.1007/s11265-019-01476-3.
12. Yu X, Tang L, Mi J, Liu J, Long L. Fault tolerant and quality of service aware routing algorithm based on priority technique for scalable network on chip architectures. *Sci Rep.* 2025;15:36578. doi:10.1038/s41598-025-20381-3.
13. Khan ZA, Abbasi U, Kim SW. An efficient algorithm for mapping deep learning applications on the NoC architecture. *Appl Sci.* 2022;12(6):3163. doi:10.3390/app12063163.
14. de Barros JB, Sampaio RC, Llanos CH. An adaptive discrete particle swarm optimization for mapping real-time applications onto network-on-a-chip based MPSoCs. In: *Proceedings of the 32nd Symposium on Integrated Circuits and Systems Design*; 2019 Aug 26–30; São Paulo, Brazil. p. 1–6. doi:10.1145/3338852.3339835.

15. Amin W, Hussain F, Anjum S, Saleem S, Baloch NK, Zikria YB, et al. Efficient application mapping approach based on grey wolf optimization for network on chip. *J Netw Comput Appl.* 2023;219(1):103729. doi:10.1016/j.jnca.2023.103729.
16. Fang J, Cai H, Lv X. Hybrid optimization algorithm based on double particle swarm in 3D NoC mapping. *Micromachines.* 2023;14(3):628. doi:10.3390/mi14030628.
17. Sahu PK, Shah T, Manna K, Chattopadhyay S. Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization. *IEEE Trans VLSI Syst.* 2014;22(2):300–12. doi:10.1109/tvlsi.2013.2240708.
18. Sadesh S, Chandrasekaran G, Thangaraj R, Kumar NS. Test scheduling of network-on-chip using hybrid WOA-GWO algorithm. *Intell Data Anal Int J.* 2025;29(3):769–86. doi:10.3233/ida-240878.
19. Li K, Shao J, Song Y. ET: a metaheuristic optimization algorithm for task mapping in network-on-chip. *Electronics.* 2025;14(14):2846. doi:10.3390/electronics14142846.
20. Jagadheesh S, Bhanu PV, Soumya J. NoC application mapping optimization using reinforcement learning. *ACM Trans Des Autom Electron Syst.* 2022;27(6):1–16. doi:10.1145/3510381.
21. Reza MF. Reinforcement learning based dynamic link configuration for energy-efficient NoC. In: *Proceedings of the 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS); 2020 Aug 9–12; Springfield, MA, USA.* p. 468–73. doi:10.1109/mwscas48704.2020.9184490.
22. Weng X, Liu Y, Xu C, Lin X, Zhan L, Wang S, et al. A machine learning mapping algorithm for NoC optimization. *Symmetry.* 2023;15(3):593. doi:10.3390/sym15030593.
23. Li Y, Zhou P. Fast and accurate NoC latency estimation for application-specific traffics via machine learning. *IEEE Trans Circuits Syst II.* 2023;70(9):3569–73. doi:10.1109/tcsii.2023.3258700.
24. Li X, Fan W, Zhang H, Ji J, Cheng T, Li S, et al. TTNNM: thermal- and traffic-aware neural network mapping on 3D-NoC-based accelerator. In: *Proceedings of the Great Lakes Symposium on VLSI 2024; 2024 Jun 12–14; Clearwater, FL, USA.* p. 364–9.
25. Mosayyebzadeh A, Amiraski MM, Hessabi S. Thermal and power aware task mapping on 3D Network on Chip. *Comput Electr Eng.* 2016;51:157–67. doi:10.1016/j.compeleceng.2015.12.001.
26. Cui Y, Zhang W, Chaturvedi V, Liu W, He B. Thermal-aware task scheduling for 3D-network-on-chip: a bottom to top scheme. *J Circuits Syst Comput.* 2016;25(1):1640003. doi:10.1142/s021812661640003x.
27. Li B, Wang X, Singh AK, Mak T. On runtime communication- and thermal-aware application mapping in 3D NoC. In: *Proceedings of the Eleventh IEEE/ACM International Symposium on Networks-on-Chip; 2017 Oct 19–20; Seoul, Republic of Korea.* p. 1–8. doi:10.1145/3130218.3130228.
28. Reza MF. Application-architecture mapping for energy-efficiency and high-performance in network-on-chip-based manycore architectures. In: *Proceedings of the 2024 International VLSI Symposium on Technology, Systems and Applications (VLSI TSA); 2024 Apr 22–25; HsinChu, Taiwan.* p. 1–2. doi:10.1109/vlsitsa60681.2024.10546359.
29. Tang J, Hong J. Reinforcement learning-driven task migration for effective temperature management in 3D noc systems. *Sci Rep.* 2025;15(1):11933. doi:10.1038/s41598-025-96335-6.
30. Samala J, Mekala BB, Soumya J. Enhancing fault-tolerant application mapping in network-on-chip through transformer network based reinforcement learning approach. *Discov Comput.* 2025;28(1):235. doi:10.1007/s10791-025-09704-0.
31. Yadav CM, Reddy BNK. Machine learning-driven fault-tolerant core mapping in Network-on-Chip architectures for advanced computing networks. *Parallel Comput.* 2026;127:103167. doi:10.1016/j.parco.2025.103167.
32. Kadri N, Chenine A, Laib Z, Koudil M. Reliability-aware intelligent mapping based on reinforcement learning for networks-on-chips. *J Supercomput.* 2022;78(16):18153–88. doi:10.1007/s11227-022-04590-5.
33. Asadi Y. A comprehensive study and holistic review of empowering network-on-chip application mapping through machine learning techniques. *Discov Electron.* 2024;1(1):22. doi:10.1007/s44291-024-00027-w.
34. Zafar A, Farooq M, Samad A. Application mapping approaches and methodologies in real-time network-on-chip (NoC) architectures. In: *Proceedings of the 2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC); 2024 Dec 18–20; Solan, India.* p. 810–5. doi:10.1109/pdgc64653.2024.10984033.

35. Khan MS, Durrani YA. A review of optimization techniques in network-on-chip (NoC) architecture. In: Proceedings of the 2024 International Conference on Engineering & Computing Technologies (ICECT); 2024 May 23; Islamabad, Pakistan. p. 1–6. doi:10.1109/icect61618.2024.10581134.
36. Kadri N, Koudil M. A survey on fault-tolerant application mapping techniques for Network-on-Chip. *J Syst Archit*. 2019;92:39–52. doi:10.1016/j.sysarc.2018.10.001.
37. Saleem S, Hussain F, Amin W, Ahmed R, Zikria YB, Ishmanov F, et al. A survey on dynamic application mapping approaches for real-time network-on-chip-based platforms. *IEEE Access*. 2023;11(1):122694–721. doi:10.1109/access.2023.3329233.
38. Amin W, Hussain F, Anjum S, Khan S, Baloch NK, Nain Z, et al. Performance evaluation of application mapping approaches for network-on-chip designs. *IEEE Access*. 2020;8:63607–31. doi:10.1109/access.2020.2982675.
39. Bose A, Ghosal P. The CTH network: an NoC platform for scalable and energy efficient application mapping solution. *IEEE Trans Nanotechnol*. 2023;22:58–69. doi:10.1109/tnano.2023.3237128.
40. Yerima WY, Ikechukwu OM, Dang KN, Ben Abdallah A. Fault-tolerant spiking neural network mapping algorithm and architecture to 3D-NoC-based neuromorphic systems. *IEEE Access*. 2023;11:52429–43. doi:10.1109/access.2023.3278802.
41. Maatar M, Dang KN, Ben Abdallah A. Cluster-based thermal-aware mapping for 3D-NoC-based NASH neuro-morphic system. In: Proceedings of the 2024 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET); 2024 Apr 27–29; Hammamet, Tunisia. p. 1–6. doi:10.1109/ic_aset61847.2024.10596243.
42. Springer T, Eiroa-Lledo E, Stevens E, Linstead E. On-device deep learning inference for system-on-chip (SoC) architectures. *Electronics*. 2021;10(6):689. doi:10.3390/electronics10060689.
43. Mirhoseini A, Goldie A, Yazgan M, Jiang JW, Songhori E, Wang S, et al. A graph placement methodology for fast chipdesign. *Nature*. 2021;594(7862):207–12. doi:10.1038/s41586-021-03544-w.
44. Alimi IA, Patel RK, Aboderin O, Abdalla AM, Gbadamosi RA, Muga NJ, et al. Network-on-chip topologies: potentials, technical challenges, recent advances and research direction. In: Network-on-chip—architecture, optimization, and design explorations. London, UK: IntechOpen; 2021. doi:10.5772/intechopen.97262.
45. Kaleem M, Isnin IF. Thermal-aware directional and adaptive routing algorithm for 3D network-on-chip. *Indones J Electr Eng Comput Sci*. 2022;27(2):1051. doi:10.11591/ijeecs.v27i2.pp1051-1061.
46. Kaleem M, Isnin I. Interval based transaction record keeping mechanism for adaptive 3D network-on-chip routing. *Int J Intell Eng Syst*. 2022;15:509–19. doi:10.22266/ijies2022.0831.46.
47. Kaleem M, Isnin IFB. Thermal-aware dynamic weighted adaptive routing algorithm for 3D network-on-chip. *Int J Adv Comput Sci Appl*. 2021;12(11):342–8. doi:10.14569/ijacsa.2021.0121139.
48. Kaleem M, Isnin IFB. A survey on network on chip routing algorithms criteria. In: Saeed F, Al-Hadhrani T, Mohammed F, Mohammed E, editors. *Advances on smart and soft computing*. Vol. 1188. Singapore: Springer; 2020. p. 455–66. doi:10.1007/978-981-15-6048-4_40.
49. John JT, Dahare N, Chaithanya B, Reuben J. Communication centric floorplanning of NoC based system on chip. *Procedia Comput Sci*. 2016;93:259–68. doi:10.1016/j.procs.2016.07.209.
50. Chao CH, Jheng KY, Wang HY, Wu JC, Wu AY. 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; 2010 May 3–6; Grenoble, France. p. 223–30. doi:10.1109/nocs.2010.32.
51. Kumar AS, Naresh Kumar Reddy B. An efficient real-time embedded application mapping for NoC based multiprocessor system on chip. *Wirel Pers Commun*. 2023;128(4):2937–52. doi:10.1007/s11277-022-10080-x.
52. Mandal SK, Krishnakumar A, Ogras UY. Energy-efficient networks-on-chip architectures: design and run-time optimization. In: Mishra P, Charles S, editors. *Network-on-chip security and privacy*. Cham, Switzerland: Springer International Publishing; 2021. p. 55–75. doi:10.1007/978-3-030-69131-8_3.
53. Spieck J, Wildermann S, Teich J. On transferring application mapping knowledge between differing MPSoC architectures. *IEEE Trans Comput-Aided Des Integr Circuits Syst*. 2022;41(11):4289–300. doi:10.1109/tcad.2022.3197527.

54. Liang SW, Wu GCY, Yee KC, Wang CT, Cui JJ, Yu DCH. High performance and energy efficient computing with advanced SoIC™ scaling. In: Proceedings of the 2022 IEEE 72nd Electronic Components and Technology Conference (ECTC); 2022 May 31–Jun 3; San Diego, CA, USA. p. 1090–4. doi:10.1109/ECTC51906.2022.00176.
55. Guo B, Liu H, Niu L. Network-on-chip (NoC) applications for IoT-enabled chip systems: latest designs and modern applications. *Int J Hi Spe Ele Syst*. 2025;34(3):2540027. doi:10.1142/s0129156425400270.
56. Choudhary J, Soumya J, Cenkeramaddi LR. RAMAN: reinforcement learning inspired algorithm for mapping applications onto mesh network-on-chip. In: Proceedings of the 2021 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP); 2021 Nov 4; Munich, Germany. p. 52–8. doi:10.1109/slip52707.2021.00019.
57. Shammasi M, Baharloo M, Abdollahi M, Baniasadi A. Turn-aware application mapping using reinforcement learning in power gating-enabled network on chip. In: Proceedings of the 2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc); 2022 Dec 19–22; Penang, Malaysia. p. 345–52. doi:10.1109/mcsoc57363.2022.00061.
58. Rao N, Ramachandran A, Shah A. MLNoC: a machine learning based approach to NoC design. In: Proceedings of the 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD); 2018 Sep 24–27; Lyon, France. p. 1–8. doi:10.1109/cahpc.2018.8645914.
59. Das S, Doppa JR, Pande PP, Chakrabarty K. Energy-efficient and reliable 3D network-on-chip (NoC): architectures and optimization algorithms. In: Proceedings of the 35th International Conference on Computer-Aided Design; 2016 Nov 7–10; Austin, TX, USA. p. 1–6. doi:10.1145/2966986.2980096.
60. Ben-Itzhak Y, Zahavi E, Cidon I, Kolodny A. HNOCS: modular open-source simulator for Heterogeneous NoCs. In: Proceedings of the 2012 International Conference on Embedded Computer Systems (SAMOS); 2012 Jul 16–19; Samos, Greece. p. 51–7. doi:10.1109/samos.2012.6404157.
61. Alagarsamy A, Gopalakrishnan L. SAT: a new application mapping method for power optimization in 2D—NoC. In: Proceedings of the 2016 20th International Symposium on VLSI Design and Test (VDAT); 2016 May 24–27; Guwahati, India. p. 1–6. doi:10.1109/isvdat.2016.8064880.
62. Alagarsamy A, Gopalakrishnan L, Ko SB. KBMA: a knowledge-based multi-objective application mapping approach for 3D NoC. *IET Comput Digit Tech*. 2019;13(4):324–34. doi:10.1049/iet-cdt.2018.5055.
63. Alagarsamy A, Mahilmaran S. A case study on cluster based application mapping method for power optimization in 2D NoC. In: Machine learning and data science. Hoboken, NJ, USA: John Wiley & Sons Ltd.; 2022. p. 89–108. doi:10.1002/9781119776499.ch6.
64. Liu J, Li Z, Li S. Noc architecture study with DFG model. In: Proceedings of the 2010 First International Conference on Pervasive Computing, Signal Processing and Applications; 2010 Sep 17–19; Harbin, China. p. 903–6. doi:10.1109/pcspa.2010.223.
65. Vemulapalli G, Pulivarthy P. Integrating Green infrastructure with AI-driven dynamic workload optimization: focus on network and chip design. In: Integrating blue-green infrastructure into urban development. Hershey, PA, USA: IGI Global Scientific Publishing; 2025. p. 397–422. doi:10.4018/979-8-3693-8069-7.ch018.
66. Cakin A, Dilek S, Tosun S. Energy-aware application mapping methods for mesh-based hybrid wireless network-on-chips. *J Supercomput*. 2024;80(11):15582–612. doi:10.1007/s11227-024-06062-4.
67. Chen Y, Zhu W, Lu Z. Travel time-based task mapping for NoC-based DNN accelerator. In: Carro L, Regazzoni F, Pilato C, editors. Embedded computer systems: architectures, modeling, and simulation. Cham, Switzerland: Springer Nature; 2025. p. 76–92. doi:10.1007/978-3-031-78377-7_6.
68. Zhu W, Chen Y, Lu Z. NoCDAS: a cycle-accurate NoC-based deep neural network accelerator simulator. *ACM Trans Model Comput Simul*. 2025;35(4):1–24. doi:10.1145/3729169.
69. Al-Hchaimi AAJ, Al-Shareeda MA, Azar AT, El-Shafai W. Reinforcement guided genetic algorithm for application mapping in network-on-chip architectures: toward transparent and efficient MPSoC scheduling. *IEEE Access*. 2025;13:177520–36. doi:10.1109/access.2025.3616179.
70. Lit A, Suhaili S, Kipli K, Rajae N. Performance analysis of NoC and WiNoC in multicore system architectures. *Int J Netw Distrib Comput*. 2025;13(1):13. doi:10.1007/s44227-024-00046-9.

71. Cui K, Rao S, Xu S, Huang Y, Cai X, Huang Z, et al. Spectral convolutional neural network chip for in-sensor edge computing of incoherent natural light. *Nat Commun.* 2025;16(1):81. doi:10.1038/s41467-024-55558-3.
72. Alagarsamy A, Mahilmaran S, Gopalakrishnan L, Ko SB. SaHNoC: an optimal energy efficient hybrid networks-on-chip architecture. *J Supercomput.* 2023;79(6):6538–59. doi:10.1007/s11227-022-04910-9.
73. Irabor E, Musavi M, Das A, Abadal S. Exploring the potential of wireless-enabled multi-chip AI accelerators. *arXiv:2501.17567.* 2025. doi:10.48550/arxiv.2501.17567.
74. Fischer T, Rogenmoser M, Benz T, Gürkaynak FK, Benini L. FlooNoC: a 645-Gb/s/link 0.15-pJ/B/hop open-source NoC with wide physical links and end-to-end AXI4 parallel multistream support. *IEEE Trans VLSI Syst.* 2025;33(4):1094–107. doi:10.1109/tvlsi.2025.3527225.
75. Reza MF. Deep reinforcement learning enabled self-configurable networks-on-chip for high-performance and energy-efficient computing systems. *IEEE Access.* 2022;10:65339–54. doi:10.1109/access.2022.3182500.
76. Choudhary J, Sudarsan CS, Soumya J. A performance-centric ML-based multi-application mapping technique for regular Network-on-Chip. *Mem Mater Devices Circuits Syst.* 2023;4:100059. doi:10.1016/j.memori.2023.100059.
77. Jain A, Kumar Dwivedi R, Alshazly H, Kumar A, Bourouis S, Kaur M. Design and simulation of ring network-on-chip for different configured nodes. *Comput Mater Contin.* 2022;71(2):4085–100. doi:10.32604/cmc.2022.023017.
78. Jain A, Kumar A, Prakash Shukla A, Alshazly H, Elmannai H, Algarni AD, et al. Smart communication using 2D and 3D mesh network-on-chip. *Intell Autom Soft Comput.* 2022;34(3):2007–21. doi:10.32604/iasc.2022.024770.
79. Joardar BK, Doppa JR, Pande PP, Marculescu D, Marculescu R. Hybrid on-chip communication architectures for heterogeneous manycore systems. In: *Proceedings of the International Conference on Computer-Aided Design*; 2018 Nov 5–8; San Diego, CA, USA. p. 1–6. doi:10.1145/3240765.3243480.
80. Madhini M, Tapas Bapu BR. Hybrid artificial humming bird and coati optimization algorithm fostered power aware application mapping in 3D-NoC system. *Wirel Netw.* 2025;31(4):3517–31. doi:10.1007/s11276-025-03937-z.
81. Ji J, Gao H, Zhang H, Lu Y, He K, Song Y, et al. NAME: NoC-based accelerators mapping exploration for high performance DNN inference. In: *Proceedings of the 2025 IEEE International Symposium on Circuits and Systems (ISCAS)*; 2025 May 25–28; London, UK. p. 1–5. doi:10.1109/iscas56072.2025.11043634.
82. Jiang Q, Xin X, Yao L, Chen B. METSM: multiobjective energy-efficient task scheduling model for an edge heterogeneous multiprocessor system. *Future Gener Comput Syst.* 2024;152:207–23. doi:10.1016/j.future.2023.10.024.
83. Kiran KA, Jacob J. Energy-aware application mapping onto 3D mesh-based network-on-chip using heuristic mapping algorithms. *J Univers Comput Sci.* 2025;31(2):136–58. doi:10.3897/jucs.123539.
84. Gaffour K, Benhaoua MK, Benyamina AH, Zahaf HE. A new congestion-aware routing algorithm in network-on-chip: 2D and 3D comparison. *Int J Comput Appl.* 2023;45(1):27–35. doi:10.1080/1206212x.2019.1679529.
85. Liu W, Yang L, Jiang W, Feng L, Guan N, Zhang W, et al. Thermal-aware task mapping on dynamically reconfigurable network-on-chip based multiprocessor system-on-chip. *IEEE Trans Comput.* 2018;67(12):1818–34. doi:10.1109/tc.2018.2844365.
86. Elgammal MA, Mohaghegh A, Shahrouz SG, Mahmoudi F, Koşar F, Talaei K, et al. VTR 9: open-source CAD for fabric and beyond FPGA architecture exploration. *ACM Trans Reconfig Technol Syst.* 2025;18(3):1–53. doi:10.1145/3734798.
87. Masdari M, Qasem SN, Pai HT. Optimizing Network-on-Chip using metaheuristic algorithms: a comprehensive survey. *Microprocess Microsyst.* 2023;103:104970. doi:10.1016/j.micpro.2023.104970.
88. Gomes de Araujo Rocha HM, Schneider Beck AC, Eduardo Kreutz M, Diniz Monteiro Maia SM, Magalhães Pereira M. Using evolutionary metaheuristics to solve the mapping and routing problem in networks on chip. *Des Autom Embed Syst.* 2023;27(1):51–83. doi:10.1007/s10617-023-09269-5.
89. Bougherara M, Rafik A, Kemcha R. Application mapping onto network on chip using particle swarm optimisation with genetic algorithm “GAPSO”. In: *Proceedings of the 2022 International Conference on Computer and Applications (ICCA)*; 2022 Dec 20–22; Cairo, Egypt. p. 1–6. doi:10.1109/icca56443.2022.10039542.
90. Yadav S, Laxmi V, Kapoor H, Gaur MS, Kumar A. Adaptive distribution of control messages for improving bandwidth utilization in multiple NoC. *J Supercomput.* 2023;79(15):17208–46. doi:10.1007/s11227-023-05208-0.

91. Rzaev ER, Monakhova EA, Romanov AY. New methods of synthesis of optimal circulant graphs for network-on-chip design. *Lobachevskii J Math.* 2024;45(12):6581–93. doi:10.1134/s1995080224606313.
92. Romanov AY. Development of routing algorithms in networks-on-chip based on ring circulant topologies. *Heliyon.* 2019;5(4):e01516. doi:10.1016/j.heliyon.2019.e01516.
93. da Silva Oliveira S, de Carvalho BM, Kreutz ME. Network-on-chip irregular topology optimization for real-time and non-real-time applications. *Micromachines.* 2021;12(10):1196. doi:10.3390/mi12101196.
94. Ali J, Maqsood T, Khalid N, Madani SA. Communication and aging aware application mapping for multicore based edge computing servers. *Clust Comput.* 2023;26(1):223–35. doi:10.1007/s10586-022-03588-1.
95. Kullu P, Yigit-Sert S, Ozkan-Okay M, Ar Y. An artificial bee colony based mapping method for three dimensional network-on-chip. In: *Proceedings of the 2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*; 2023 Aug 16–18; Windhoek, Namibia. p. 1–6. doi:10.1109/etncc59188.2023.10284966.
96. Dehghani A, Fadaei S, Ravaei B, RahimiZadeh K. Deadline-aware and energy-efficient dynamic task mapping and scheduling for multicore systems based on wireless network-on-chip. *IEEE Trans Emerg Topics Comput.* 2023;11(4):1031–44. doi:10.1109/tetc.2023.3315298.
97. Reza MF, McCloud Z. Heuristics-enabled high-performance application mapping in network-on-chip based multicore systems. In: *Proceedings of the 2023 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*; 2023 Jul 23–25; Berlin, Germany. p. 1–6. doi:10.1109/coins57856.2023.10189228.
98. Patra R, Maji P, Raj Y, Mondal HK. Enhancing predictive accuracy using machine learning for network-on-chip performance modeling. *Comput Electr Eng.* 2025;123:110041. doi:10.1016/j.compeleceng.2024.110041.
99. Mo L, Li X, Kritikakou A, Zhai X. Contention and reliability-aware energy efficiency task mapping on NoC-based MPSoCs. *IEEE Trans Rel.* 2025;74(1):2010–26. doi:10.1109/tr.2024.3377732.
100. Liu Y, Ye Y. An efficient branch-and-bound routing algorithm for optical NoCs. In: *Proceedings of the 2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*; 2024 Jan 22–25; Incheon, Republic of Korea. p. 860–5. doi:10.1109/asp-dac58780.2024.10473882.
101. Dadmand MR, Farazkish R, Reza A, Rafiei Nazari R, Faghieh Mirzaee R. Adaptive multi-beltway thermal-aware routing algorithm for 3D NoC system. *J Supercomput.* 2025;81(10):1095. doi:10.1007/s11227-025-07580-5.
102. Reddy BNK, Rahman MZU, Lay-Ekuakille A. Enhancing reliability and energy efficiency in many-core processors through fault-tolerant network-on-chip. *IEEE Trans Netw Serv Manage.* 2024;21(5):5049–62. doi:10.1109/tnsm.2024.3394886.
103. Reddy BNK, James A, Kumar AS. Fault-tolerant core mapping for NoC based architectures with improved performance and energy efficiency. In: *Proceedings of the 2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*; 2022 Oct 24–26; Glasgow, UK. p. 1–4. doi:10.1109/icecs202256217.2022.9970825.
104. Bose A, Ghosal P. Scalable Run-time task mapping on BNoC topology focusing on energy-efficiency. In: *Proceedings of the 2022 15th IEEE/ACM International Workshop on Network on Chip Architectures (NoCArc)*; 2022 Oct 2; Chicago, IL, USA. p. 1–6. doi:10.1109/nocarc57472.2022.9911366.
105. Bienia C, Kumar S, Singh JP, Li K. The PARSEC benchmark suite: characterization and architectural implications. In: *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*; 2008 Oct 25–29; Toronto, ON, Canada. p. 72–81. doi:10.1145/1454115.1454128.
106. Bougherara M, Nedjah N, Bennouar D, de Macedo Mourelle L. Routing in 3D NoCs using genetic algorithm and particle swarm optimization. In: *Computational Science and Its Applications—ICCSA 2023 Workshops*; 2023 Jul 3–6; Athens, Greece. Switzerland, Cham: Springer Nature; 2023. p. 601–13. doi:10.1007/978-3-031-37105-9_40.
107. Muhsen YR, Husin NA, Zolkepli MB, Manshor N, Al-Hchaimi AAJ, Ridha HM. Enhancing NoC-based MPSoC performance: a predictive approach with ANN and guaranteed convergence arithmetic optimization algorithm. *IEEE Access.* 2023;11:90143–57. doi:10.1109/access.2023.3305669.
108. Spieck J, Wildermann S, Teich J. A scenario-based DVFS-aware hybrid application mapping methodology for MPSoCs. *ACM Trans Des Autom Electron Syst.* 2024;29(4):1–43. doi:10.1145/3660633.
109. Akinyele D. Machine learning-enhanced data routing in next-generation SoC-NoC systems. *SSRN J.* 2024. doi:10.2139/ssrn.4995487.

110. Yin J, Sethumurugan S, Eckert Y, Patel C, Smith A, Morton E, et al. Experiences with ML-driven design: a NoC case study. In: Proceedings of the 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA); 2020 Feb 22–26; San Diego, CA, USA. p. 637–48. doi:10.1109/hpca47549.2020.00058.
111. Anantharajaiah N, Xu Y, Lesniak F, Harbaum T, Becker J. DREAM: distributed reinforcement learning enabled adaptive mixed-critical NoC. In: Proceedings of the 2023 IEEE Computer Society Annual Symposium on VLSI (ISVLSI); 2023 Jun 20–23; Foz do Iguacu, Brazil. p. 1–6. doi:10.1109/isvlsi59464.2023.10238569.
112. Saleem S, Hussain F, Baloch NK. IWO-IGA—a hybrid whale optimization algorithm featuring improved genetic characteristics for mapping real-time applications onto 2D network on chip. *Algorithms*. 2024;17(3):115. doi:10.3390/a17030115.
113. Zhang B, Gu H, Zhang GL, Yang Y, Ma Z, Schlichtmann U. A 3D hybrid optical-electrical NoC using novel mapping strategy based DCNN dataflow acceleration. *IEEE Trans Parallel Distrib Syst*. 2024;35(7):1139–54. doi:10.1109/tpds.2024.3394747.
114. Reza MF, Yeazel A. Mapping model and heuristics for accelerating deep neural networks and for energy-efficient networks-on-chip. In: Proceedings of the SoutheastCon 2024; 2024 Mar 15–24; Atlanta, GA, USA. p. 119–26. doi:10.1109/southeastcon52093.2024.10500232.
115. Li S, Zhou S, Xue Y, Fan W, Cheng T, Ji J, et al. HAS-RL: a hierarchical approximate scheme optimized with reinforcement learning for NoC-based NN accelerators. *IEEE Trans Circuits Syst I*. 2024;71(4):1863–75. doi:10.1109/tcsi.2024.3359912.
116. Ramesh S, Manna K, Gogineni VC, Chattopadhyay S, Mahapatra S. Congestion-aware vertical link placement and application mapping onto 3-D network-on-chip architectures. *IEEE Trans Comput-Aided Des Integr Circuits Syst*. 2024;43(8):2249–62. doi:10.1109/tcad.2024.3371255.
117. Fan W, Li S, Zhu L, Lu Z, Li L, Fu Y. Communication synchronization-aware arbitration policy in NoC-based DNN accelerators. *IEEE Trans Circuits Syst II*. 2024;71(10):4521–5. doi:10.1109/tcsii.2024.3395054.
118. Kang Z, Zhu J, Xiao X, Li S, Wang L, Ma D, et al. LSM-based hotspot prediction and hotspot-aware routing in NoC-based neuromorphic processor. *IEEE Trans VLSI Syst*. 2024;32(7):1239–52. doi:10.1109/tvlsi.2024.3370850.
119. Bougherara M, Amara R. Application mapping onto network on chip using genetic algorithm and ant colony optimisation. In: Proceedings of the 2023 International Conference on Computer and Applications (ICCA); 2023 Nov 28–30; Cairo, Egypt. p. 1–6. doi:10.1109/icca59364.2023.10401486.
120. Fernandes R, Marcon C, Cataldo R, Sepulveda J. Using smart routing for secure and dependable NoC-based MPSoCs. *IEEE/ACM Trans Netw*. 2020;28(3):1158–71. doi:10.1109/tnet.2020.2979372.
121. Khodadadi E, Barekatain B, Yaghoubi E, Mogharrabi-Rad Z. FT-PDC: an enhanced hybrid congestion-aware fault-tolerant routing technique based on path diversity for 3D NoC. *J Supercomput*. 2022;78(1):523–58. doi:10.1007/s11227-021-03906-1.
122. Jiao J, Shen R, Chen L, Liu J, Han D. RLARA: a TSV-aware reinforcement learning assisted fault-tolerant routing algorithm for 3D network-on-chip. *Electronics*. 2023;12(23):4867. doi:10.3390/electronics12234867.
123. Jagadheesh S, Bhanu PV, Soumya J, Cenkeramaddi LR. Reinforcement learning based fault-tolerant routing algorithm for mesh based NoC and its FPGA implementation. *IEEE Access*. 2022;10:44724–37. doi:10.1109/access.2022.3168992.
124. Lee SC, Han TH. Q-function-based traffic- and thermal-aware adaptive routing for 3D network-on-chip. *Electronics*. 2020;9(3):392. doi:10.3390/electronics9030392.
125. Zhiri MA, Krichene H, Sandionigi C, Pillement S. HT-NoC: reconfigurable high throughput network-on-chip for AI dataflow accelerators. In: Giorgi R, Stojilović M, Stroobandt D, Jiménez PB, Barros AB, editors. *Applied reconfigurable computing. Architectures, tools, and applications*. Cham, Switzerland: Springer Nature; 2025. p. 15–32. doi:10.1007/978-3-031-87995-1_2.
126. Abdollahi M, Mohammadi S. Insertion loss-aware application mapping onto the optical cube-connected cycles architecture. *Comput Electr Eng*. 2020;82(1):106559. doi:10.1016/j.compeleceng.2020.106559.
127. Sambangi R, Manghnani H, Chattopadhyay S. LPNet: a DNN based latency prediction technique for application mapping in Network-on-Chip design. *Microprocess Microsyst*. 2021;87(1):104370. doi:10.1016/j.micpro.2021.104370.

128. Swain K, Das AS, Misra S, Singh V, Das T, Choudhury S. A comprehensive study on graphs, design process, and mapping techniques based on evolutionary algorithm for network on chip architecture. In: *Applied soft computing techniques*. Waretown, NJ, USA: Apple Academic Press; 2025. doi:10.1201/9781003593133-11.
129. Koenen M, Doan NAV, Wild T, Herkersdorf A. Exploring task and channel mapping strategies in fail-operational and hard real-time NoCs. In: *Proceedings of the 2020 IEEE Nordic Circuits and Systems Conference (NorCAS)*; 2020 Oct 27–28; Oslo, Norway. p. 1–7. doi:10.1109/norcass51424.2020.9264993.
130. Celik C, Bazlamacci CF. Effect of application mapping on network-on-chip performance. In: *Proceedings of the 2012 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*; 2012 Feb 15–17; Munich, Germany. p. 465–72. doi:10.1109/pdp.2012.48.
131. Mo L, Zhang J, Al-Hasan TM, Cui M, Zhai X, Gao Q, et al. Reliable and energy optimized task mapping for heterogeneous multicore NoC based on partial task duplication and multipath routing. *IEEE Internet Things J*. 2025;12(23):51240–53. doi:10.1109/jiot.2025.3612945.
132. Kumar AS, Rao TVKH. Scalable benchmark synthesis for performance evaluation of NoC core mapping. *Microprocess Microsyst*. 2020;79(1):103272. doi:10.1016/j.micpro.2020.103272.
133. Weber II, Zanini VB, Moraes FG. Reinforcement learning for thermal and reliability management in manycore systems. *Des Autom Embed Syst*. 2024;29(1):2. doi:10.1007/s10617-024-09292-0.
134. Kumar AS, Hanumantha Rao TVK, Kumar Reddy BN. Exact formulas for fault aware core mapping on NoC reliability. In: *Proceedings of the 2020 IEEE 17th India Council International Conference (INDICON)*; 2020 Dec 10–13; New Delhi, India. p. 1–5. doi:10.1109/indicon49873.2020.9342427.
135. Liu X, Zhou X, Fu C, Yang D, Ji S, Zhang N, et al. Fault modeling, testing, and repair for chiplet interconnects. *Chip*. 2025;46(1):100160. doi:10.1016/j.chip.2025.100160.
136. Fogli A, Zhao B, Pietzuch P, Giceva J. CHARM: chiplet heterogeneity-aware runtime mapping system. New York, NY, USA: ACM; 2026.
137. Mallya NB, Goel B, Sourdis I. MEMPLEX: a memory system with replication and migration of data for multi-chiplet NUMA Architectures. In: *Proceedings of the 39th ACM International Conference on Supercomputing*; 2025 Jun 8–11; Salt Lake City, UT, USA. p. 1219–33. doi:10.1145/3721145.3725776.
138. Ibáñez Bolado M, Pérez Pavón B, Luis Bosque Orero J. REX: a remote execution model for continuous scalability in multi-chiplet-module GPUs. *Future Gener Comput Syst*. 2026;178:108268. doi:10.1016/j.future.2025.108268.
139. Lv M, Dong H, Fan W. Efficient fault tolerance and diagnosis mechanism for Network-on-Chips. *J Netw Comput Appl*. 2025;237:104133. doi:10.1016/j.jnca.2025.104133.
140. Al-Hchaimi AAJ, Muhsen YR, Pamucar D, Simic V. A security-aware multi-criteria decision-making framework for ordering task mapping techniques in 3D-NoC based MPSoC architectures of IoT. *Comput Stand Interfaces*. 2026;96:104075. doi:10.1016/j.csi.2025.104075.
141. Al-Hchaimi AAJ, Muhsen YR, Gwad WH, Alkayal ES, Al Ogaili RRN, Alyasseri ZAA, et al. Prioritizing network-on-chip routers for countermeasure techniques against flooding denial-of-service attacks: a fuzzy multi-criteria decision-making approach. *Comput Model Eng Sci*. 2025;142(3):2661–89. doi:10.32604/cmesci.2025.061318.
142. Shen S, Bonato T, Hu Z, Jordan P, Chen T, Hoefler T. ATLAHS: an application-centric network simulator toolchain for AI, HPC, and distributed storage. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*; 2025 Nov 16–21; Louis, MO, USA. p. 349–67. doi:10.1145/3712285.3759838.
143. Xiong G, Luo X, Liu W. Coherence-aware task graph modeling for realistic application. *arXiv:2509.09094*. 2025.