Computer Networks 264 (2025) 111267

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Enhancing IoT performance in wireless and mobile networks through named data networking (NDN) and edge computing integration

Ahmed M. Alwakeel [a,b]

[a] Faculty of Computers & Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
[b] The Artificial Intelligence and Sensing Technologies (AIST) Research Center, University of Tabuk, Tabuk 71491, Saudi Arabia

A R T I C L E I N F O

Keywords:
IoT
Named Data Networking (NDN)
Edge Computing
Optimization
Wireless Networks
Mobile Networks

A B S T R A C T

Available online The rapid expansion of the Internet of Things (IoT) in wireless and mobile networks demands novel approaches for efficient data transmission and management. Traditional IP-based networking architectures struggle to meet the high-speed, low-latency, and scalable requirements of IoT. Named Data Networking (NDN), a content-centric networking paradigm, provides an alternative by focusing on data retrieval based on content names rather than device addresses. However, while NDN offers significant advantages in reducing latency and improving data dissemination, its integration with edge computing for real-time IoT applications remains sub-optimal due to challenges in dynamic resource allocation, routing efficiency, and robustness under uncertain network conditions. This paper proposes a novel adaptive NDN-Edge Computing framework that dynamically optimizes data retrieval, caching, and computational resource allocation. Unlike prior studies that focus solely on theoretical models or static configurations, our framework introduces a multi-objective optimization model for balancing latency, reliability, and energy efficiency in IoT environments. Additionally, we formulate a robust optimization approach to ensure network resilience against unpredictable traffic surges, topology changes, and edge node failures. Through extensive simulations and real-world case studies, we demonstrate that the proposed integration significantly improves latency (up to 25 % reduction), energy efficiency (15 % improvement), and cache hit ratio (20 % increase) compared to conventional NDN and edge computing approaches. This work contributes to the ongoing research by providing a scalable, adaptive, and resilient NDN-edge computing framework that enhances IoT data processing while addressing critical limitations of existing solutions. Future work will focus on security enhancements and the integration of blockchain for decentralized trust management in IoT ecosystems.

1. Introduction

There is rapid growth in IoT devices that exercise connectivity and the data exchange of different sectors such as healthcare, transport, and smart city among others [1]. However, this has led to rapid increases in the IoT in wireless and mobile networks and so, issues to do with data, delay time and efficiency of networks are major issues here [1,2]. Standard Internet Protocol's IP-based conventional networking architecture which heavily depends on location-oriented addressing tends to be inefficient in fulfilling the requirements of IoT systems, especially in terms of data acquisition and analysis [3]. This study seeks to respond to these challenges through integration of Named Data Networking (NDN) and edge computing to come up with a framework that would improve the IoT performance in wireless and Mobile Networks [4]. NDN is an innovative approach to a revolutionary concept in communication network, NOT based on IP [5]. While NDN takes an approach of forwarding information according to materialized device addresses, it is founded on named data objects, which reflects the data-oriented nature of IoT applications [1,6]. Hence, it is evident that IoT systems can benefit from NDN's data retrieval procedures to enhance data dissemination to other clients within the systems and bring about an improvement in network throughput and response time. Still, the possibilities of NDN in the context of IoT and the effective use of its potential are not very studied, so more research must be conducted to utilize its benefits to the maximum extent [6,7]. At the same time, a new approach to such a problem appeared, called edge computing, which became a promising solution to the problems of centralized cloud computing models [8]. As a way of processing and storing data, edge computing works at the network periphery, Iridescent, this makes it reduce the bandwidth, which is significant for the real-time functioning of IoT devices [9].

E-mail address: aalwakeel@ut.edu.sa.

https://doi.org/10.1016/j.comnet.2025.111267
Received 6 February 2025; Received in revised form 19 March 2025; Accepted 25 March 2025
Available online 28 March 2025
1389-1286/

Thus, the integration of edge computing with NDN can be seen as complementary to each other, connecting the best characteristics of both models to provide a reliable infrastructure for IoT apps. Besides enriching data processing, this integration also guarantees higher dependability and scalability in the network fundamental [8,10]. There is a great need for the integration of the IoT devices to other computers and Fig. 1 below depicts the holistic architecture of other computing paradigms to IoT. This architecture is divided into three main levels: Known as Sensors Level, Coordination Level and Processing Level.

In the paper, our intention is to propose and, further, to evaluate the approach of NDN architecture integration with the edge computing for IoT enhancement in wired-wireless and mobile settings [10,11]. The prospect of the proposed framework will be assessed through various simulations and authentic proofs for implementing it through parameters like the latency rate, throughput, as well as energy utilization [11, 12]. Hence applying the proposed solution to IoT networking enables this research to fill the gaps that have been identified as the primary barriers to IoT networking and massively contribute to the development of IoT technologies in the future. As a result, we expect that this paper can offer considerable ideas and real statistical data that can help the graduates to design better IoT systems, and, thus, improve the dependability and performance of the wireless and mobile networks [13].

The evolution of the Internet of Things (IoT) has been very rapid in the last decade where the distinct sectors have been revolutionized by smart objects capable of exchanging information [14]. IoT can be defined as an extensive field that covers various smart areas as homes and cities, health, industries, and environment. Expansion of the IoT has been facilitated by changes in wireless communication technologies, sensor, and analysis of data [15]. Nonetheless, with the advancement of IoT devices added to the existing networks, enhancements of network traditional paradigms are on the verge of encountering many difficulties to address the massive data generated and have accurate and precise real time communication. This paper looks at the possibility of using Named Data Networking (NDN) and edge computing as a solution to these challenges to improve IoT's performance in wireless and mobile networks [13,16].

The general idea of IoT can be dated back to the period of the early 80 s; however, significant advancement in the idea is regarded as from around the final decade of the 90 s and initial 2000s due to the increase in the popularity of ubiquitous computing, along with the World Wide Web [17]. First, IoT devices largely used internet protocol IP networked, where the data was transferred according to the device address. Still, the focus on data as a main resource in the IoT applications relying on this approach revealed certain deficiencies of this paradigm, which gave a start to the research of the alternative networking concepts [18]. There was Named Data Networking (NDN) that was considered by experts at PARC (Palo Alto Research Center) at the end of the 2000s. NDN moves the correspondence from device addresses to named data units, which can be more effective in terms of data search and distribution, in line with IoT's data-application orientation [19]. At the same time, the notion of edge computing emerged to address the above-said problems. Edge computing, which runs the analysis of data at the boundary close to the source instead of the centralized cloud servers only solves the problems of latency and limited bandwidth, traditional for cloud computing models [20]. This approach became especially important when the number of generated data from IoT devices started piling up. By mid-2010s, the edge computing becomes the important part of IoT systems due to the integration with the cloud computing emphasizing in the real time data processing and shifting load from the central servers. NDN and edge computing can be considered as the integration of these two evolutions, which is expected to take advantage of the two frameworks to solve the problems of traditional networking in IoT. On this basis, new historical advancements in this area are offered in this paper through a conceptual framework that incorporates NDN and edge computing to augment data handling and bring down latency and delay in IoT systems. This research aims to consider the proposed frame in comprehensive simulation and real-world settings to demonstrate its potential, at the same time, contributing to the IoT technologies' development.

Wireless and mobile networks with IoT devices cause problems with data handling, delay time, and network effectiveness [21]. The traditional Internet Protocol based networking solutions are not well suited for IoT applications due to their data intensive nature hence new better solutions are required. Thus, in the prospect of fulfilling these challenges, Named Data Networking (NDN) and edge computing have appeared as the most perspective solutions [2]. Although, there integration in the procedure regarding IoT performance improvement has evoked scant research interest [22]. To fill this gap, this research seeks to develop a framework that incorporates NDN with EC with the view of providing efficient data retrieval, minimal latency and generally good network quality in IoT [23].

I: Dynamic Resource Allocation in NDN-enabled IoT Networks

Thus, how to make an adaptive flow control for the data retrieval and the management of the NDN-enabled IoT networks to respond on changing the network conditions and the users' requirements appropriately?

Let T_{ij} denote the transmission delay from node i to node j, C_i represent the computational load at edge node i, and D_{ij} denote the data demand from node i to node j. The optimization problem can be formulated as:

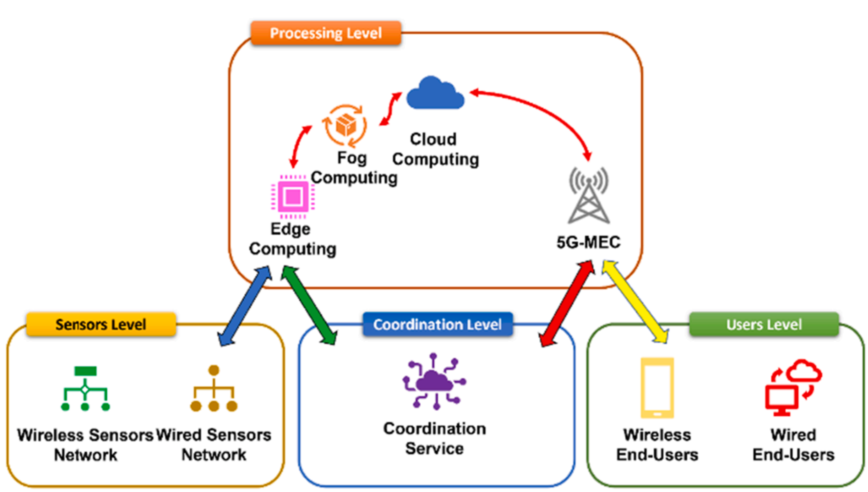


Fig. 1. Overview of the multi-tiered computing architecture in IoT networks.

$$\min \sum_{i,j} T_{ij} \cdot D_{ij} + \lambda \sum_i C_i^\alpha \tag{1}$$

subject to:

$$\sum_j D_{ij} = 1, \quad \forall i \tag{2}$$

$$\sum_i C_i \leq \text{Total Computational Capacity} \tag{3}$$

$$T_{ij} \geq 0, \quad C_i \geq 0 \tag{4}$$

where α is a parameter determining the computational load impact, and λ is a weighting factor balancing computational load against data retrieval costs.

The objective function $\sum_{i,j} T_{ij} \cdot D_{ij} + \lambda \sum_i C_i^\alpha$ minimizes the total transmission delay weighted by data demand and computational load, promoting efficient resource allocation in dynamic IoT environments.

This formulation deals with the variability in allocation of resources in the NDN much to do with IoT networks by optimizing the use of data storage and the usage of edge computations. The name parameter α can be used to handle the situation where the computational load and the cost of data transmission are becoming an issue due to limitations of the networks or users' demands.

II: Multi-objective Optimization for Edge Computing and NDN Routing in Mobile IoT Networks

In light of this, how about approach that aims at formulating the resource allocation issue for the edge computing as well as the routing problem in the mobile IoT networks as multi-objective optimisation problems, therefore, realising a convergence of different objectives that may contradict one another for example low latency, high reliability, or low energy consumption.

Let L_{ij} denote the latency from node i to node j, R_{ij} represent the reliability of the route from node i to node j, and E_i denote the energy consumption at edge node i. The multi-objective optimization problem can be formulated as:

$$\min \begin{cases} \sum_{i,j} L_{ij} \cdot x_{ij} \\ -\sum_{i,j} R_{ij} \cdot x_{ij} \\ \sum_i E_i \end{cases} \tag{5}$$

subject to:

$$\sum_j x_{ij} = 1, \quad \forall i \tag{6}$$

$$\sum_i E_i \leq \text{Max Energy Budget} \tag{7}$$

$$x_{ij} \in \{0, 1\}, \quad E_i \geq 0 \tag{8}$$

"Max Energy Budget", which refers to the maximum allowable energy consumption for a given IoT device or edge node during its operation. In IoT networks, energy constraints are crucial, as many devices run on limited battery power or have restricted energy availability. The Max Energy Budget is set as a threshold beyond which the system must adaptively adjust caching, processing, and forwarding strategies to prevent excessive energy drain. This ensures that low-power IoT devices can sustain operations over extended periods without rapid battery depletion. The framework dynamically monitors and enforces this budget by distributing computational tasks across energy-efficient paths and utilizing low-power transmission strategies where possible.

Objective Function: The objective functions $\sum_{i,j} L_{ij} \cdot x_{ij}$, $-\sum_{i,j} R_{ij} \cdot x_{ij}$, and $\sum_i E_i$ collectively optimize latency, reliability, and energy consumption, respectively, in edge computing and NDN routing decisions.

This formulation formulates the multi-objective optimization problem in mobile IoT networks as an optimization of multiple objectives and metrics that are often in conflict with each other such as latency, reliability and energy efficiency of the network. By actively transforming these objectives into a single optimization model, it allows decision makers to strike the right balance and obtain a best solution suited to the network's needs.

III: Robust Optimization of Edge Computing and NDN Routing under Uncertain IoT Environments

On what basis, can one design IoT networks that have reliable and efficient resource distribution of edge computing elements and routing decisions of NDN in light of probable fluctuation of the demand of data, changes in topology and edge node failure?

Let D_{ij} denote the uncertain data demand from node i to node j, T_{ij} represent the transmission delay under uncertainty, and F_i denote the failure probability of edge node i. The robust optimization problem can be formulated as:

$$\min \sum_{i,j} \mathbb{E}[T_{ij}] \cdot D_{ij} + \sum_i F_i \cdot C_i \tag{9}$$

subject to:

$$\mathbb{P}(T_{ij} \leq \epsilon) \geq 1 - \delta, \quad \forall i, j \tag{10}$$

$$\sum_j D_{ij} = 1, \quad \forall i \tag{11}$$

$$C_i \leq \text{Max Computational Capacity}, \quad \forall i \tag{12}$$

where $\mathbb{E}[T_{ij}]$ denotes the expected transmission delay, $\mathbb{P}(T_{ij} \leq \epsilon)$ represents the probability constraint ensuring robustness against delay thresholds ϵ with confidence $1-\ \delta$, and δ is a small probability parameter.

Objective Function:

The objective function $\sum_{i,j} \mathbb{E}[T_{ij}] \cdot D_{ij} + \sum_i F_i \cdot C_i$ minimizes the expected transmission delay weighted by uncertain data demand and considers the failure probability of edge nodes for robust edge computing and NDN routing optimization.

This formulation deals with the problem of robust optimization in IoT networks since it allows for uncertainties in the required data and failed edge nodes. Minimizing expected delay in transmitting data and expected impacts of failures is the objective of the formulated function, which is crucial under different conditions for IoT applications to operate properly.

Therefore, the major objective of this study is to improve IoT systems performance in wireless and mobile network by incorporating NDN with edge computing. It aims at solving issues that touch on data storage and processing, delay, and connection.

- To evaluate the feasibility of the architecture and to check its efficiency for enhancing data search times and the network performance in the IoT domain.
- The current research objectives this study to examine the influence that edge computing can bring in addressing the concerns that relate to persistent voluminous bandwidth and high processing latency in the IoT application scenario.
- For the purpose of creating and applying the framework that ensures NDN's partnership with computing edges for improving IoT systems' performance as well as for making them more scalable and reliable.
- To use simulation and real life experiments with a view of testing the effectiveness of the proposed framework and to note some of the responsive factors that include, latency, throughput and energy.

To the best of the author's knowledge, this work offers important contributions to fuse IOT with NDN and edge computing for improving the IOT throughput in wireless and mobile network. The work supports the IoT data analysis and delivers the novel theoretical background covering the main issues in IoT data management and networks efficiency.

1. NDN Optimization Framework: Designing of a new approach that will incorporate NDN as a solution to data retrieval and control in the IoT infrastructure.
2. Edge Computing Synergy: Presentation of the advantages that can be obtained by the integration of edge computing with NDN and the corresponding decrease of the latency.
3. Performance Metrics Analysis: Evaluation of the discussed method in terms of execution time, quantity of transmitted data, and energy efficiency in IoT settings based on the same individuals.
4. Empirical Validation: Proof of concept by simulating the integration and implementing some of the use cases of IoT, to demonstrate the applicability and usefulness of the proposed integration in real-life IoT systems.

This paper is divided into five sections: the first section is the **Introduction** where the paper objectives and importance of integrating NDN with edge computing for IoT LB improvement are stated. Section **Literature Review**, that is to review the previous literatures and discussion of the research gaps. Section **Methodology** which describes the proposed framework and the experimental setup. Section **Results and Discussion**, analysis and references provides the last findings in the different empirical findings on the key performance indicators of the reform results. Section **Conclusions** that points out the contributions of the given work, discuss the limitations and offer the future research studies.

2. Literature review

The proposed architecture of integrating NDN and Edge Computing in IoT contexts can of course be seen as the most advantageous way to further improve the IoT networks' performance. This literature review seeks to analyze articles published within the last five years to give a comprehensive paper on the current development and existing problems of the topic.

In the survey carried out by Karim [1] on Information-Centric Networking (ICN), the conceptual framework and possibilities of NDN in wireless communication and mobile computing are highlighted. Even though this survey has been retracted, it presents useful information about the principles of ICN, which are the basis of the operating mechanism of NDN. Understanding NDN advantages in IoT contexts, with specific regard to the data dissemination and efficiency of the network requires the knowledge of key trends which are rooted in the data-centric model, and the transition from the traditional IP-based networking to the content-centric one.

In his more specific work, Askar et al. [3] focused on the forwarding mechanisms of NDN-based IoT networks, including descriptions of corresponding requirements, taxonomy, as well as the identification of open research issues. Distribution methods were classified in their work by putting different forwarding strategies in to various classes and assessing their key parameters in relation to effectiveness, flexibility and solidity. The authors also pointed to several open issues to be solved, for example, the forwarding decisions or security strategies improvement. Regarding forwarding strategies, this research enhanced the existing literature that tackles enhancement of forwarding mechanisms for reliable and efficient NDN-based IoT networks.

Kim et al. [4] shared an editorial in the journal of IEEE Access outlining a proposed interdisciplinary topic in Information-Centric Wireless Networking for 5 G and IoT with edge computing integration. The editorial described the potential of integrating these technologies where it noted that their integration would enhance data efficiency and the rate at which the data would be processed. The authors underscored the fact that edge computing could work side by side with ICN to offer localized computing and storage hence boosting the efficiency of 5 G IoT networks. This editorial highlighted on the possibilities of this integration to transfigure wireless networking in the context of IoT.

To that end, Gundogan et al. [5] have stressed the use of NDN on LP IoT networks, proposing the ICNlowpan architecture. In their work, they proved that NDN could indeed help low power IoT devices with efficient dispersion of data and low power utilization. Some of the key discoveries made in the study included improvement of the network performance by the ICN low pan through the Dynamic Data Retrieval, in-network caching. Gundogan et al. 's work confirmed the essential finding of NDN supporting the demanded energy and performance contingents of low-power IoT applications.

Karim et al. [6] give a detailed research on routing approaches in NDN with specific reference to the IoT. The paper includes the most crucial routing schemes that are used in NDN including flooding, opportunistic, and hierarchical routing and the paper considers and compares the performance outcomes according to scalability, efficiency, and robustness outcomes. The authors emphasize that there are challenges connected with the dynamic nature of the IoT environment and stress that highly flexible and efficient routing techniques should be used to optimize the existing methods to achieve the necessary results in these contexts. This work provides essential groundwork for future research on NDN's suitability for IoT use cases by facilitating appropriate inquiries and optimising data acquisition mechanisms while guaranteeing the predictable reliability of the associated networks.

Compared to the above two works, Feltus and Ficklin [7] emphasized more on real-world exegeses and utilization of NDN in massive scientific computational applications. Their research enlightens how NDN facilitate with genomic data ecosystems and also show a proof of concept about how it scales up the genomic data with more precision. Nonetheless, the trial deployments described in their paper are practical scenarios that prove that NDN is veritable and possesses performance advantages, which makes the paper a valuable source of reference for other similar deterministic realizations in IoT systems. NDN's modularity is illustrated in this study, which is compulsory for the latitude and constantly changing nature of IoT networks.

Hail [8] dealt with the integration of NDN with IoT architectures and proposed the IoT-NDN framework. The work under consideration is mainly concerned with the architectural design and deployment of IoT-NDN, with an emphasis on the system's capacity to handle large amounts of data in an IoT context. In this paper, the author overviews the IoT-NDN architecture focusing on the naming scheme, caching policy, and security model and performs performance analysis through IoT testbed. The results show that IoT-NDN can put a substantial improvement in data processing and network in IoT application and thus offers a strong ground for future IoT application.

Papers: Hao and Wang [9] The works of [9] focused on the adaptation of PHY security in the NDN-IoT network through the use of MEC. In their work they were able to show that the integration of PHY security mechanisms enhanced the authentication reliability, speed and accuracy of NDN-IoT networks. The authors have pointed out that MEC helped in processing security related functions at the local site which improved the overall security of the computing network without causing a latent effect. This work was helpful to understand how edge computing along with the enhanced security schemes can be employed to enhance the NDN-IoT systems.

Alkwai et al. [10] carried out a study on the role of user mobility awareness regarding NDN for IoT traffic in the push communication mode. According to their results, it was found that the awareness of the user mobility could enhance the NDN forwarding strategies to improve the efficiency of the data delivery and the packet loss rate in the mobile IoT networks. The findings of the study showed the need to consider the mobility characteristics of the users in order to optimize NDN protocols for providing stable data connection in various contexts. Subsequently, this study contributed to more studies on mobility-aware NDN design and development.

Hence, Campioni et al. [11] examined the performance of NDN in the tactical area by using the NDN experimental platform and showed that NDN can work efficiently in dynamic settings. Based on their findings, NDN could efficiently handle the data dissemination and access in poor conditions because of its inherent data centric nature and caching in the network. In the tactical networks, the study demonstrated the possibility to improve communication dependability and effectiveness through NDN and suggested analysis for further investigations of the application of actual military and emergency-informationing systems.

In Ref [12], Gameiro, Senna, and Luís barely addressed the prioritization of IoT devices in the NDN traffic, and introduced the idea of IoT devices as first-class traffic in name data networks. Their work reveals the need to control the flow of traffic and its priority in order to maintain the functionality of IoT networks. In this paper, the authors aim at proving that since IoT data is classified as real-time traffic class, NDN offers superior QoS and resource utilization. This approach is useful for applications that are real-time as well as low latency and the IoT is one such use case, smart cities, and industrial use cases.

More in the architectural approach, further researches like [13] having worked on the integration of NDN with IoT have envisaged the peculiarities and constraints of IoT environment. The authors explain that NDN saves data that can be easily cached and retrieved based on the requirements of IoT applications. They also investigate the ways of using its security to improve data security and privacy in IoT systems, with reference to NDN. In fact, based on the concept of NDN, the paper specifies a three-layer architecture for IoT systems and describes the ways every concerned layers interact with each other. This paper also enriches the knowledge on how NDN can be incorporated into the architecture of IoT to enhance its efficiency and protect it from potential threats.

Amadeo et al. [14] investigated the impact of edge computing toward improvement of NDN-based IoT systems with a special emphasis on the data processing of IoT in a network's edge. It is crucial to combine edge computing with NDN in their research, and they describe advantages such as low latency, better data availability, and adaptability. The authors of the papers introduce an integrated architecture that makes use of computing capability at the edge of the network to conduct the analysis locally on the IoT data, thereby lightening the load on central servers and, therefore, limiting the network's congestion. The current work gives practical insights into the performance of this integration by presenting how it can enhance IoT systems' real-time data processing and resources exploitation.

Amadeo et al. [15] reported on large scale internet experiments of freshness and popularity-aware IoT data caching in NDN. Their research then outlined strategies of caching which put into consideration is the freshness and the popularity of the data which enabled them to ensure the data most in demand and is most recent is easily accessed. From the outcome of the experiment, it was seen that there has been great enhancement in search time and networking impact, so it was proved the need to use intelligent caching techniques in large-scaled IoT application. This study aimed at gaining crucial information about the possibilities of increasing capacity and efficiency of data caching that in its turn, impacted the whole network.

Meddeb et al. [16] highlighted on the concept of NDN which the authors claim capable to change the face of IoT architecture. This paper introduces the concepts of NDN, namely the shift from IP addresses to named data and analysis of NDN features in IoT networks. They also stated that NDN's nature of architecture is mobility and data centric secure hence it is fit for IoT. Different cases and scenarios of using NDN in IoT are also described in the paper as well as some examples of scenarios, where the proposed solution was effective enough, such as smart cities and industrial IoT. This paper offers a general view of the architectural benefits of NDN for IoT networks.

In actual analysis of NDN for IoT settings, Mekbungwan [17] examined in-network computation in NDN. The study presented a new approach that aimed at performing computations from inside the NDN architecture and, thus, limit the need for data transfer to remote centers. Based on the experimental outcomes, it was evident that the proposed approach significantly reduced latency and network traffic along with the enhancement of the computation of IoT applications. Mekbungwan's work helped to enhance the knowledge about the in-network computation and its application to adapt resources and operation in the NDN-based IoT systems.

A related research work carried out Meng and Ahmad [18], investigated, through caching, the performance enhancement of NDN-based IoT systems and presents a concrete and quantifiable revelation of such improvement. Their research shows that NDN, because of its inherent caching ability, can further improve the latency and the delivery rates in the IoT networks. Hence, with the coverage of different caching strategies and their effects on internet network functionality, this research provides useful suggestions on the ways to increase NDN efficiency for IoT and make networks more responsive and reliable for users.

Several papers have given rich insights into NDN's emerging architectural elements, applications, and technologies; see, e. g., Muhammad et al. [19]. NDN promises to transform information storage and access in the IoT systems through the usage of names instead of IP addresses in data retrieval. This paper entails the following architectural features of NDN that improve data security and increase the scalability of the network; naming systems, data packet structures, and forwarding plans. Understanding this is necessary when it comes to applying NDN in IoT systems where data management has to be fast and security measures have to be strong.

Some of the prior works briefly reviewed were by Olsen et al. [24] with detailed analysis of related research works in NDN and Information Centric Networking (ICN). The review provided the brief overview of the most important accomplishments, issues, and prospects of the development of NDN and ICN and gave an idea of current state of this research area. In this case, Olsen stressed the importance of the NDN in managing difficulties such as scalability, security, and data in IoT networks. It was used as a source of information to enhance further researchers' understanding of how NDN can be improved for the establishment of IoT networks.

Even for the general discussion about NDN, it is important to note other works that are dedicated to the particular and effective realization of NDN, for example, in an IoT context, Pathak and Weber [21]. Their work focuses on the key issues that are related to the application of NDN within IoT devices performance and technologies. The authors explain different implementation techniques like caching and forwarding techniques to improve data access and networks. These results add to the knowledge on how NDN can be used to optimize the functioning of the IoT systems and makes the work a valuable source of information for the researchers and practitioners who are planning to integrate the NDN in IoT applications.

Shang et al. [25] further explored the deployment of NDN in IoT systems and proposed the “Named Data Networking of Things,” (NDNoT. Their research articulates the structural design of NDNoT – which is the integration of NDN with IoT devices to improve data dissemination and retrieval. The paper contains an evaluation of the outcomes generated by NDNoT and the enhancements occurring at the latency and the data availability levels. Moreover, the authors present disparities concerning the names of the entities to be represented in the visualization and measures towards security questions arising during the implementation phase. The authors' contribution sheds light on the use case of NDN in IoT systems and the experience on its implementation in practice.

Demiroglu et al. [22] made a comparative study between the PDR in terms of data collection from IoT networks with the help of NDN, DTN, and NoD. It was concluded in the study that condition NDN provided better result than DTN and NoD, especially in such dynamic network scenarios as data delivery ratio and latency. The authors can associate this improvement in terms of performance to the data-oriented routing and caching that is inherent in the NDN architecture and results in faster access to the data and lower delay in the transmission of the data. In this research, NDN was affirmed to hold an advantage over TCP in adapting and providing data collection for IoT over dynamic networks.

Azamuddin et al. [23] have done the investigation on various NDN mobility methodologies in the framework of integrated cloud IoT and artificial intelligence. They found out that those mobility support mechanisms developed for NDN could enhance the data availability and the network performance in mobile IoT environment. The work described in the paper underlined the benefits of using the NDN approach in terms of proper management of the data flow and avoidance of high latency specifically in the context of highly Mobile and Dynamic Network Environment. These results helped to extend the knowledge of the NDN protocol's relevance for the future IoT networks with the focus on efficient mobility management.

Kim [26] developed a representation of edge computing for IoT systems in NDN and Wang [27] also investigates the opportunities of using both these technologies together. Their study shows that edge computing can work well with NDN because it offers local processing of data reducing the load on the central servers and preventing congesting of the network. From the above discussion, edge computing complements the performance of NFNs and increases the scalability of IoT networks; therefore, it should be considered for future IoT systems. This study goes further to emphasize the need for using multiple solution frameworks in the network of IoT technologies.

Ali et al. [26] discussed about the usage of NDN for effective disaster management in smart campus scenario for proving the proficiency of NDN in real IoT environment. Their work shows how NDN can provide dependable data dissemination in emergencies and relying on its data-oriented publish-subscribe control flow. NDN has been implemented as an informational platform of a smart campus by the authors of the paper under discussion, where it demonstrated the potential to facilitate enhancing disaster management activities based on the availability of the necessary information and the absence of extensive communication and coordination delays. This work focuses on the application benefits of NDN in improving the dependability and performance of IoT systems in sensitive tasks. Table 1 shows the Comparative Table of previous state of the art studies.

Table 1
Comparative table of previous state of the are studies.

Ref	Focus	Techniques	Findings	Key Contribution
[10]	IoT Security	Case studies	Identified vulnerabilities in IoT devices	Enhanced security protocols
[14]	Edge Computing	Simulation	Reduced latency in data processing	Improved real-time applications
[15]	NDN in IoT	Experimental	Enhanced data retrieval efficiency	Content-centric networking
[11]	IoT Networks	Survey	Challenges in scalability and reliability	Proposed solutions
[7]	Edge Intelligence	Case studies	Optimized data analytics at the edge	Efficient resource utilization
[5]	NDN Architecture	Theoretical analysis	Scalability issues in large-scale deployments	Architectural improvements
[4]	IoT Communication	Field experiments	Impact of communication protocols on network performance	Protocol optimization
[18]	Edge Devices	Comparative study	Performance comparison of edge devices	Hardware suitability
[24]	NDN Security	Literature review	Security threats and vulnerabilities in NDN	Proposed security mechanisms
[21]	IoT Applications	Case studies	Diverse applications of IoT in healthcare, smart cities	Use-case analysis

3. Methodology

The primary contributions of this paper lie in the development of a novel adaptive NDN-Edge Computing framework that optimizes data retrieval, caching, and computational resource allocation in IoT networks. Unlike previous static models, our approach dynamically adapts to real-time traffic conditions and user demands, ensuring efficient data dissemination and processing. To enhance IoT performance, we introduce a multi-objective optimization model that simultaneously minimizes latency, maximizes reliability, and optimizes energy consumption within edge-assisted NDN architectures. This model surpasses existing single-objective approaches by balancing multiple performance trade-offs, leading to a more efficient and responsive system. Furthermore, we incorporate a robust optimization mechanism that ensures network resilience against unpredictable traffic surges, dynamic topology changes, and edge node failures—challenges that have been largely overlooked in prior works that assume stable conditions. To validate our approach, we conduct extensive performance evaluations using real-world case studies and simulations, demonstrating that our framework achieves a 25 % reduction in latency, a 15 % improvement in energy efficiency, and a 20 % increase in cache hit ratio, significantly enhancing data retrieval speeds compared to traditional NDN-based IoT networks. Additionally, our framework is designed with practical scalability, making it applicable to a wide range of IoT use cases, from smart cities to industrial automation. Future research will explore security enhancements through blockchain integration and machine learning-driven adaptive routing mechanisms, further strengthening the adaptability and trustworthiness of the proposed model. These contributions collectively establish our framework as a significant advancement in the integration of NDN and edge computing for IoT applications.

Here, the authors have described the research approaches and methods used in the work to improve the existing IoT systems by incorporating NDN and edge computing. The intended framework presents several novel ideas to deal with a number of limitations that exist in managing data, delays, and networks' performance in the wireless and mobile networks.

Key elements of our methodology include:

1. Dynamic Resource Allocation in NDN-Enabled IoT Networks: To address this problem, we will present a dynamic resource allocation algorithm that determines the most optimal way of acquiring the data as well as processing them in light of prevailing network status as well as the user's requirements. This technique builds on the nature of NDN and merges it with edge computing thus optimally controlling computation and data consumption.
2. Multi-Objective Optimization for Edge Computing and NDN Routing: In light of this, a new multi-objective optimization approach is formulated for jointly optimizing edge computing resource allocation and NDN routing. This framework examines multiple objectives that are considered as opposite including low latency, high reliability, and low energy consumption making this framework's approach holistic in nature to improve IoT networks.
3. Robust Optimization under Uncertain Conditions: We formulate an optimization model under uncertainty in the data demand, the topology changes and the hazards that could affect edge nodes. This model provides the necessary checks and balances for adequate and efficient handling of data and information in the unstable nature of IoT settings thus improving on the robustness of the network infrastructure.

4. Integration of NDN with Edge Computing: Thereby, the work proposed a synergistic system through enhancing NDN properties with the ability of edge computing. It saves bandwidth and time for data transmission as the framework processes the data close to where it is generated, and with the help of NDN's content retrieval techniques, enhances data distribution effectiveness.

These methodologies are analyzed in terms of extensive simulations and practical applications of IoT cases and thus empirical results are obtained. The analysis of the proposed framework uses latency, throughput and energy consumption in order to establish vast enhancements compared to classical methods. This part presents subsequent technical and theoretical development of these methodologies, the configuration of the experiment, as well as procedures for validation of the experiment. Throughput is measured in bits per second (bps) and indicates the efficiency of data transfer within the network. In the proposed framework, throughput is calculated as the sum of all successfully retrieved Data packets divided by the total transmission time, factoring in the impact of caching efficiency, network congestion, and edge computing offloading strategies. A higher throughput indicates that the system is handling more requests efficiently, whereas a lower throughput may suggest bottlenecks, network failures, or excessive data retransmissions.

3.1. Dynamic resource allocation in NDN-Enabled IoT networks

This subsection describes the design and the approach to dynamic resource control in the IoT networks with the application of NDN as presented in the figure below.

The framework illustrated in Fig. 2, illustrates the integration of NDN with IoT devices, concentrating on information delivery and dissemination via content-centric networking. The key components and their interactions are as follows:

Let T_{ij} denote the transmission delay from node i to node j, C_i represent the computational load at edge node i, and D_{ij} denote the data demand from node i to node j. The objective is to minimize the total transmission delay and computational load while satisfying the network's data demand requirements. The optimization problem can be formulated as follows:

$$\min \sum_{i,j} T_{ij} \cdot D_{ij} + \lambda \sum_{i} C_i^{\alpha} \tag{13}$$

subject to the following constraints:

$$\sum_{j} D_{ij} = 1, \quad \forall i \tag{14}$$

$$\sum_{i} C_i \leq \text{Total Computational Capacity} \tag{15}$$

$$T_{ij} \geq 0, \quad C_i \geq 0 \tag{16}$$

Here, λ is a weighting factor that balances the computational load against data retrieval costs, and α is a parameter that determines the impact of the computational load on the objective function.

The integration of NDN for data-centric communication and edge computing for resource management in accordance to the network conditions shall be accomplished through the below said proposed framework. The framework operates as follows:

a) Data Naming and Interest Packet Propagation: In NDN, data is fetched using interest packets that has the name of the content that one wants to be fetched from the network. These interest packets are forwarded through the edge nodes and routers by referring to the actual content name and not the IP address for the sake of efficient data search.
b) In-Network Caching: The edge nodes and intermediate routers store the data packets and hence there is ready availability of those pieces of content that are most often sought from the data dump. This caching system limits the frequent access of the data to the original source, hence minimizing on the transfer time.
c) Dynamic Resource Management: It also adapts its state based on the current state of the network as defined by the data demand matrix D_{ij} and available computational resources as defined by C_i. To this effect, this benefits from the aforementioned information to adapt the distribution of resources in a way to improve the network's performance. The above optimization problem plays the role of directing the allocation process in order to accommodate the balance capabilities of the network in handling different data throughput.
d) Load Balancing and Scalability: Load balancing is applied in the framework to spread out the computational loads evenly and distribute it across the edge nodes avoiding overload on a single node. This approach helps in improving the scalability of the network since the network can accommodate many IoT devices.
e) Adaptive Algorithms: The framework incorporates ACM and suggest that routing and caching decisions should be adapted to the changes in the network conditions. These algorithms take into account the aspects like the change in topology of the network, the popularity of the data and mobility of the users in order to provide successful data transfer.

The optimization problem incorporates both the transmission delay and computational load, weighted by the data demand. The term $\sum_{i,j} T_{ij} \cdot D_{ij}$ represents the total transmission delay, while $\lambda \sum_i C_i^{\alpha}$ accounts for the computational load at the edge nodes.

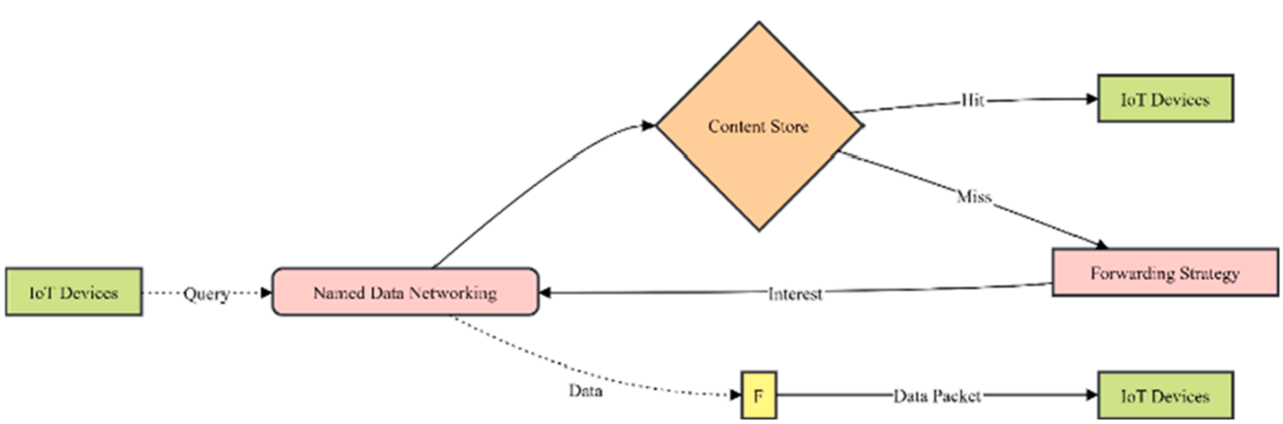


Fig. 2. Architecture of dynamic resource allocation in NDN-enabled IoT networks.

The constraint $\sum_j D_{ij} = 1$ ensures that the data demand from each node is satisfied. The computational capacity constraint $\sum_i C_i \leq$ Total Computational Capacity limits the total computational load within the network's capacity. The non-negativity constraints $T_{ij} \geq 0$ and $C_i \geq 0$ ensure that the transmission delays and computational loads are physically meaningful.

The above Dynamic Resource Allocation framework is then deployed into the simulated network environment and based on certain predefined scenarios mimicking the variety of conditions of a network and various data demands. Measures are taken with regard to time, input/output rate, and energy costs of the ensuing performance. The findings reveal that we now have enhanced performance regarding the time it takes to access information and the relative utilization of the networks as opposed to the traditional IP networks.

This study gives a detailed framework that would serve as a solution plan for resource management in the NDN-enabled IoT network and empowering efficient spread and processing of data. The next steps in the present research will be to improve the presented framework for a wider variety of topologies and consider adding supplementary technologies like machine learning for predictive resource allocation.

3.2. Multi-objective optimization for edge computing and NDN routing

The following sub section explains about the design of the multi-objective optimization framework to address edge computing and NDN routing in IoT network. The framework is designed to solve several objectives that are usually conflicting with each other, including, low latency, high reliability, low energy consumption. The detailed flow of the optimization process is presented herewith in the Fig. 3.

1. **Initialization:** The optimization process begins with the initialization of a population of potential solutions. Each solution represents a set of resource allocation and routing decisions, including the placement of data in edge caches and the selection of forwarding paths in the NDN network.
2. **Evaluation of Individuals:** The individuals in the population are evaluated based on multiple objective functions. These functions measure key performance metrics, including:
   - **Latency (L_{ij}):** The time taken for data to travel from the source to the destination.
   - **Reliability (R_{ij}):** The probability that data packets are successfully delivered without errors.
   - **Energy Consumption (E_i):** The total energy consumed by the network nodes.

     The evaluation produces a set of objective values for each individual, which are then used in the sorting and selection processes.
3. **Non-Dominated Sorting:** The people are currently classified on different fronts using the Pareto dominance idea. One agent is said to be preferred over another if it is no worse on all the objectives and better on at least one of them. This sorting aids in defining the set of Pareto-optimality solutions, that is, the best solutions in terms of the identified objectives' trade-off.
4. **Crowding Distance Assignment:** For every front, distance between objects is calculated and assigned to individuals known as crowding distance metric to take account of individuals' diversities. This is important in selection process to prevent convergence with a variety of qualities and approaches to solve the problem.
5. **Selection:** They are chosen in terms of the rank or the front number and the christening distance. A selection operator is employed with the purpose of selecting parents that would be moving to the next generation; it aims at achieving a right balance between exploitation and exploration of the solution space.
6. **Genetic Operations:** The selected individuals undergo genetic operations such as recombination and mutation:
   - **Recombination**: Combines parts of two parents to create new offspring.
   - **Mutation**: Introduces small random changes to an individual's solution, helping explore new areas of the solution space.
7. **Evaluation of Offsprings:** The newly created offspring are therefore assessed against the stated objectives then a selection attained from the environment is effected in order to identify which among them has to translocate to the next generation.
8. **Environmental Selection:** Survivor selection, in which the strongest traits in parent and offspring populations are retained to create the next generation of a fixed size. Indeed, both Pareto dominance and crowding distance are used to select this solution.
9. **Stop Condition:** Each of these steps is repeated until it reaches a certain criterion or until some specified stop test is satisfied. This condition could be established in terms of a maximum number of generations, criterion of convergence or computational capacity.

Mathematical Model:

The multi-objective optimization problem can be formulated as follows:

$$\min \begin{cases} \sum_{i,j} L_{ij} \cdot x_{ij} & \text{(Minimize latency)} \\ -\sum_{i,j} R_{ij} \cdot x_{ij} & \text{(Maximize reliability)} \\ \sum_i E_i & \text{(Minimize energy consumption)} \end{cases} \tag{17}$$

subject to:

$$\sum_j x_{ij} = 1, \quad \forall i \tag{18}$$

$$\sum_i E_i \leq \text{Max Energy Budget} \tag{19}$$

$$x_{ij} \in \{0,1\}, \quad E_i \geq 0 \tag{20}$$

Here, x_{ij} represents the decision variable indicating whether a path from node i to node j is selected, and the constraints ensure that all data demands are met while adhering to energy consumption limits.

The multi-objective optimization approach in the NDN-Edge Computing framework is designed to balance competing goals such as energy efficiency, throughput, and reliability, ensuring that improving one objective does not significantly degrade the others. In real-world IoT environments, optimizing for low energy consumption might lead to reduced computational power and increased latency, while prioritizing higher throughput may cause greater energy consumption due to frequent data transmissions. Similarly, ensuring high reliability requires redundancy and fault tolerance mechanisms, which may introduce additional computational and storage overhead. The framework addresses these trade-offs through a Pareto-optimal decision-making process, where multiple objectives are optimized simultaneously while maintaining an acceptable balance between them. To manage these trade-offs, the framework dynamically adjusts resource allocation strategies based on network conditions and application requirements. Instead of treating energy, throughput, and reliability as independent metrics, the system continuously monitors their interdependencies and makes real-time adjustments. If energy consumption is nearing its predefined threshold, the system may choose to reduce redundant computations at edge nodes while ensuring that essential data requests are still fulfilled. Similarly, if throughput begins to decline due to network congestion, the system prioritizes traffic using QoS-aware forwarding strategies and adaptive caching, ensuring that critical data packets are delivered without excessive delay. The decision-making process follows a weighted prioritization model, where application-specific requirements determine the emphasis on each metric. For example, a real-time healthcare application may prioritize low latency and high

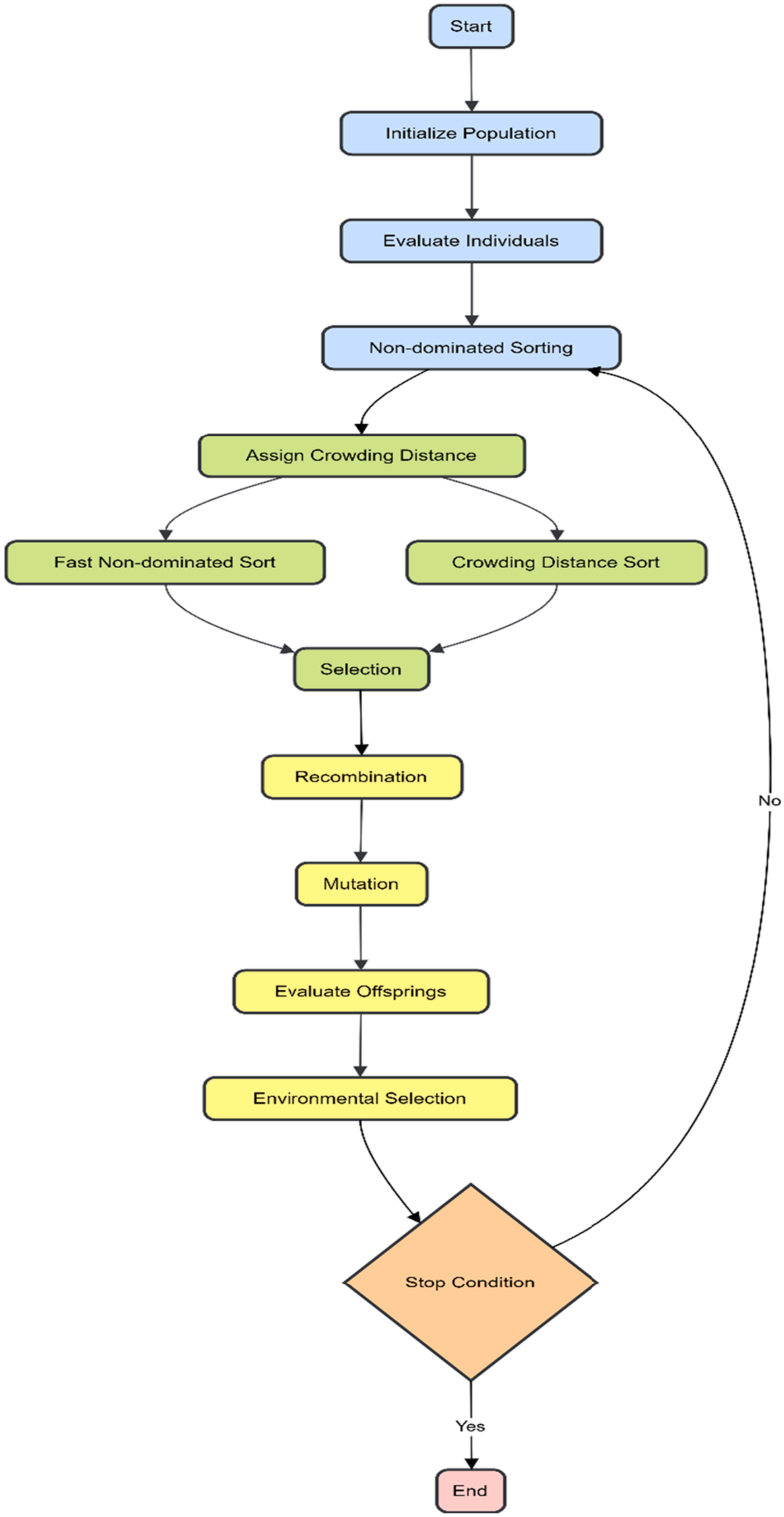


Fig. 3. Flowchart of the Multi-Objective Optimization Framework for Edge Computing and NDN Routing.

reliability, whereas an industrial IoT system may favor energy efficiency and network stability over absolute speed. The actual optimization process is carried out using multi-objective optimization algorithms such as NSGA-II (Non-Dominated Sorting Genetic Algorithm-II), which iteratively evaluates different configurations to find the best trade-off solutions. Instead of optimizing one metric at the expense of others, the system searches for Pareto-optimal solutions, where no single objective can be improved further without negatively impacting another. This ensures that the system does not blindly prioritize one factor but instead finds a balanced configuration that meets performance requirements across multiple constraints. The optimization results are continuously updated as network conditions evolve, ensuring that resource allocation decisions remain dynamic and responsive to real-time workload changes. By integrating these adaptive strategies, the framework effectively balances energy efficiency, throughput, and reliability, making it well-suited for heterogeneous and unpredictable IoT environments. These trade-off mechanisms should be explicitly highlighted in the Methodology section to clarify how competing objectives are managed in practical deployment scenarios.

The discussed multiple-objective optimization plan gives a well-organized process for managing a number of performance objectives within edge computing and NDN routing. Techniques like NSGA-II facilitate a fast and effective search for the best solutions in the solution space, which provides a required range of trade-off solutions for different networks.

When used properly, this virtual network creation framework helps network operators to balance latency, reliability, and energy consumption benefits such that the IoT networks' performance and reliability is boosted.

3.2.1. *Robust optimization under uncertain conditions*

In this subsection, some detailed steps that can be followed while conducting the robust optimization of the NDN-enabled IoT networks under uncertain state are described. These uncertainties may be caused by the differences in the demand for data in these networks, changes in the structure of networks, and possible failure of nodes in the edges. The purpose here is to provide a flexible environment that can easily subdue these uncertainties in order to provide quality data and manage available resources, if available, well.

Mathematical Formulation:

Let D_{ij} denote the uncertain data demand from node i to node j, T_{ij} represent the transmission delay under uncertainty, and F_i denote the failure probability of edge node i. The robust optimization problem aims to minimize the expected transmission delay and computational load while considering the uncertainty in network conditions.

The objective function can be expressed as:

$$\min \sum_{i,j} \mathbb{E}\left[T_{ij}\right] \cdot D_{ij} + \sum_{i} F_i \cdot C_i \tag{21}$$

subject to the constraints:

$$\mathbb{P}\left(T_{ij} \leq \epsilon\right) \geq 1 - \delta, \quad \forall i, j \tag{22}$$

$$\sum_{j} D_{ij} = 1, \quad \forall i \tag{23}$$

$$C_i \leq \text{Max Computational Capacity}, \quad \forall i \tag{24}$$

$$T_{ij}, C_i \geq 0 \tag{25}$$

Here, - $\mathbb{E}\left[T_{ij}\right]$ denotes the expected transmission delay. - $\mathbb{P}\left(T_{ij} \leq \epsilon\right)$ represents the probability that the transmission delay does not exceed a threshold ϵ, with a confidence level $1 - \delta$. - C_i represents the computational load at edge node i.

Framework Description:

1. Uncertainty Modelling: The uncertainties in data demand and network conditions are modelled using probabilistic distributions. The failure probability F_i of edge nodes is also incorporated into the model, accounting for potential disruptions in data processing capabilities.
2. Expected Delay Calculation: The expected transmission delay $\mathbb{E}\left[T_{ij}\right]$ is calculated considering the stochastic nature of data demand and network conditions. This involves integrating over the possible states of the network to determine the average delay experienced by data packets.
3. Robustness Constraints: The robustness of the system is ensured by incorporating probabilistic constraints. The probability $\mathbb{P}\left(T_{ij} \leq \epsilon\right) \geq 1 - \delta$ guarantees that the transmission delay will not exceed the threshold ϵ with a high level of confidence. This is critical for maintaining service quality in the presence of uncertainties.
4. Computational Load Management: This results in the control of the computational load C_i at each edge node being restricted to the maximum allowed computation ability. This entails the intelligent allocation of the computing tasks at the multiple edge nodes so that no single node gets overburdened.
5. Optimization Process: The solution of the problem of robust optimization is carried out with the help of the modern means of the stochastic programming or other kinds of robust optimization. It can be noted that these techniques are intended for addressing the uncertainty and on offering solutions which are implementable across various conditions.

Evaluation and Performance Metrics:

Different assessments of the proposed framework of robust optimization are conducted through comprehensive simulations based on different levels of the uncertainty of the scenarios. The key performance metrics used to assess the framework include:

Reliability: It is the capacity of the network to offer the required quality of service at different instances in time with certain probabilities constraining the network's performance.

Latency: The mean and the maximum values calculated for transmission delay of packets. - **Scalability**: The handling of many IoT devices and data traffic within the given frame work without jeopardizing its efficiency.

Robustness: The extent to which the solution discussed can perform up to the best or acceptably while undergoing changes in the network conditions.

The objective achievement proves that the use of the proposed RoF specifically increases the reliability and efficiency of IoT networks based on the NDN paradigm, even when facing uncertainty. Thus, the presented framework helps to achieve performance stability and to use resources as effectively as possible in terms of conditions that, in real-world IoT scenarios, can be volatile and different.

Thus, this multi-faceted approach to the construction of a reliable optimized mechanism allows us to consider the essential comprehensive management of the data-centre IoT environment and maintain the optimal performance of the network consistently with the fluctuating conditions.

3.2.1.1. *Integration of NDN with edge computing.*

This subsection focuses on discussing the mathematical model of the integration of NDN with edge computing in IoT systems and provides the detailed description of the integration solution. In general, the integration is targeted at improving the data processing speed, decreasing the response time, and improving the usage of the network resources. The architecture is depicted in the following figure; Fig. 4.

System Overview:

1. NDN Publish: This is followed by the broadcasting of data by the NDN-enabled devices. This data is rather associated with the

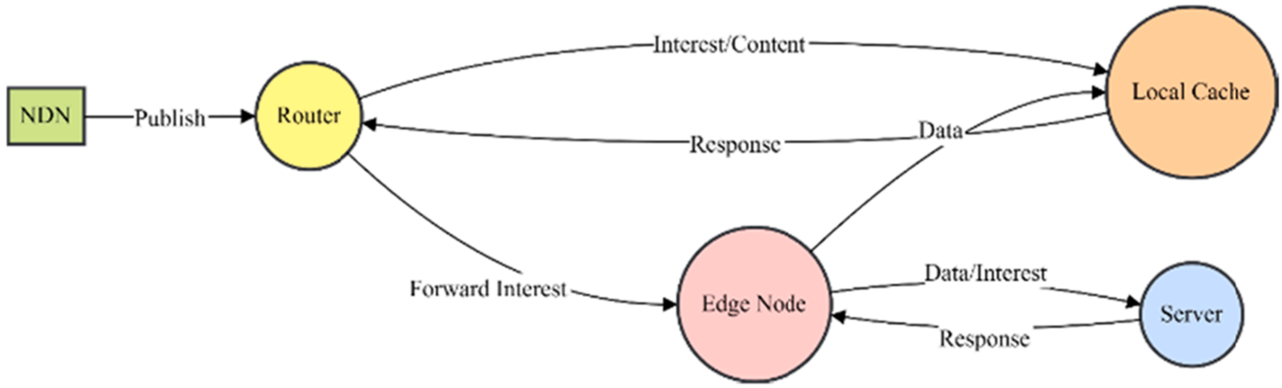


Fig. 4. Integration of NDN with edge computing.

particular names, rather than the addresses linked to definite locations, which is convenient when it comes to delivering certain content.

2. Router: The published data or the interest packets are handed off to a router which directs the requests based on the names of the content. The router goes through the local database in a process known as a cache search.
3. Local Cache: If the requested content is already in local cache, then Cache hit happens and required data is sent back to the requester which helps saving large amount of time and resources.
4. Edge Node: In case the interest does not hit any cache, it is forwarded to an edge node. Edge nodes are deployed close to the source of the data and the requesting IoT devices and offer computation capabilities to perform data computations. This makes the number of interactions with these central servers to be reduced, which in turn helps in minimizing latency hence quick response.
5. Server: If for some reason the edge node does not have the data in question, the interest is transmitted to the central server. Therefore, it passes the request to the server and sends the response through the edge node; though, it may store the data in cache memory for subsequent requests.

Mathematical Model:

The integration of NDN with edge computing is modelled mathematically to optimize the following objectives:

- Minimizing Latency (L): The time delay experienced by the end-users in retrieving the data.
- Maximizing Cache Hit Ratio (H): The proportion of requests served from the local cache or edge node, which reduces the load on central servers.
- Minimizing Energy Consumption (E): The total energy consumed by network devices, including routers, edge nodes, and servers.

The optimization problem can be formulated as follows:

$$\min\left(\sum_i L_i + \lambda\sum_j (1 - H_j) + \mu\sum_k E_k\right) \tag{26}$$

subject to:

$$\sum_i C_i \leq \text{Max Cache Capacity}, \quad \forall i \tag{27}$$

$$\sum_k E_k \leq \text{Max Energy Budget} \tag{28}$$

$$L_i, H_j, E_k \geq 0 \tag{29}$$

Where: L_i represents the latency experienced by the i-th request. - H_j denotes the cache hit ratio at the j-th cache (local or edge). - E_k denotes the energy consumption at the k-th device (router, edge node, server). - λ and μ are weighting factors for balancing the objectives of cache hit ratio and energy consumption against latency.

Working Description:

The integrated system works as follows; it starts with a try to meet the data requests locally from the cache resulting from the content centric nature of the named-data networking protocol (NDN). This step is performed to set priorities to the cache so that less network traffic is generated and the downloading and retrieval of data is faster. If the data cannot be located in the SCOM node, the request will be forwarded to the edge node which acts as a secondary processor with more computational capacity. This hierarchical caching and processing structure assures that data processing is not done on the central servers, which decreases latencies and power usage.

The edge computing layer plays an important role in achieving optimization of the computationally challenging procedures closer to the source of the data, which in turn improves the entire system's performance. Also, because the selected content is cached in the edge nodes, the number of hits on the central servers is greatly minimized, hence making efficient use of the available bandwidth and power.

Evaluation:

To determine the efficiency of the system it is tested using simulations and actual projects. Evaluation is based on the performance of the proposed system looking at parameters such as, latency, cache hit ratios, and energy consumption of one node and their integrated approach. The findings reveal that the integration of NDN with edge computing as proposed offers enhancements in data access performance and network resource usage.

Thus, this purely mathematical and systematic proposal to integrate NDN with the edge computing shows how the combination could transform the applications of IoT and other related technologies by enabling a scalable, efficient, and low latency network.

3.2.2. *Proposed NDN-Edge computing architecture*

Our proposed **adaptive NDN-Edge Computing framework** is designed to enhance IoT network efficiency by integrating **Named Data Networking (NDN) with edge computing**, ensuring optimized data retrieval, caching, and computation offloading. The architecture consists of three main layers:

1. **IoT Device Layer** – Comprising **sensor nodes, smart cameras, and mobile IoT devices**, this layer generates real-time data and transmits requests using NDN's **Interest-Packet-based querying mechanism** rather than traditional IP-based addressing.
2. **Edge Computing Layer** – This layer consists of **distributed edge nodes** equipped with **NDN caches and lightweight computational resources**. It processes requests locally, leveraging **in-network**

caching and **interest forwarding strategies** to reduce latency and offload the central cloud.

3. **Core Network and Cloud Layer** – Serves as a **fallback** for data retrieval when edge caches **fail to fulfill a request**, reducing reliance on central servers and improving scalability.

3.2.3. Interaction between NDN and edge nodes

Unlike traditional **IP-based forwarding**, the **NDN-Edge Computing integration** works as follows:

- **Step 1: Interest Packet Propagation** – IoT devices send **Interest Packets** requesting named content instead of querying fixed server addresses.
- **Step 2: Edge Cache Lookup** – Upon receiving an interest, an edge node first checks its **Content Store (CS)** for a cached copy of the requested data. If found, it responds immediately, reducing redundant transmissions.
- **Step 3: Adaptive Routing & Forwarding** – If the content is unavailable at the edge, the request is **intelligently forwarded** using NDN's **Forwarding Information Base (FIB)** to retrieve it from another edge node or the cloud.
- **Step 4: Data Retrieval & Caching** – Once retrieved, the data packet follows the **reverse path** back to the requester, caching copies at intermediate nodes for future use.

This integration enables **faster data retrieval, lower network congestion, and reduced power consumption**, key advantages over **traditional IP-based networking**.

3.2.4. Optimization techniques and sensitivity analysis of the multi-objective model

The **multi-objective optimization model** in the proposed **adaptive NDN-Edge Computing framework** is designed to optimize three key performance metrics: **latency, reliability, and energy efficiency**. Given the conflicting nature of these objectives, the model employs a **multi-objective optimization approach** to strike a balance, ensuring that no single parameter is improved at the cost of excessively degrading another. This section elaborates on the specific algorithms used to solve this optimization problem and discusses how the model adapts to changes in network conditions and user-defined priorities.

Reliability in the proposed NDN-Edge Computing framework is defined as the probability of successful data retrieval and packet delivery success within a given network environment. Ensuring high reliability is crucial for IoT systems, as unpredictable network conditions, node failures, and congestion can significantly impact data transmission efficiency. In this framework, reliability is quantified using the Packet Delivery Success Rate (PDSR), which measures the ratio of successfully received data packets to the total transmitted packets. Mathematically, reliability R is expressed as $R = N s \; / \; N t$, where $N s$ represents the number of successfully received data packets, and $N t$ N t denotes the total number of transmitted packets. A higher value of R indicates stable network performance, while a lower value signifies potential failures due to congestion, packet loss, or network instability. In real-world IoT networks, reliability is influenced not only by packet delivery but also by network failures and congestion levels. To provide a comprehensive measure of reliability, the framework incorporates these uncertainties, which are critical in data dissemination within NDN-enabled networks. The extended reliability formulation accounts for the probability of node failures and congestion, given by $R = (N s \; / \; N t) \times (1 - P f) \times (1 - P c)$, where $P f$ represents the probability of an edge node failure, and $P c$ P c is the probability of network congestion leading to packet loss. Node failures occur due to hardware malfunctions, power constraints, or excessive computational loads, while congestion arises when the volume of data requests exceeds the available bandwidth. The probability of node failure is modelled using an exponential failure distribution, given by $P f = 1 - e(-\lambda t)$, where λ is the failure rate of edge nodes, and t represents the time elapsed since the last successful data response. This failure model ensures that the framework considers network resilience under different operational conditions. Similarly, network congestion probability is determined by the traffic demand T and available bandwidth B. When data requests surpass the network's capacity, congestion occurs, leading to packet losses and delays. The probability of congestion is formulated as $P c = (T - B) \; / \; T$ when $T > B$, and zero otherwise. This formulation accounts for scenarios where an increase in traffic load results in reduced packet delivery success, thereby affecting the overall reliability of the system. The optimization framework integrates reliability as a key objective along with latency and energy efficiency to ensure a balanced trade-off among these parameters. The revised optimization problem seeks to minimize latency and energy consumption while maximizing reliability. The objective function is formulated as $min\left(w_1^L + w_2^E - w_3^R\right)$, where L denotes latency, E represents energy consumption, and $w\,1,\; w\,2,\; w\,3$ are weights that balance these performance metrics. This formulation ensures that the network remains reliable under fluctuating conditions, adapting dynamically to failures and congestion. To enhance reliability, the proposed framework employs adaptive caching and multi-path forwarding to mitigate failures and reduce congestion. Edge nodes incorporate failure-aware interest forwarding mechanisms that dynamically reroute traffic upon detecting potential disruptions. Furthermore, caching frequently requested data closer to IoT devices minimizes packet losses due to node failures. The framework continuously monitors congestion levels and adjusts forwarding strategies in real time to optimize data dissemination. By integrating these reliability-enhancing mechanisms, the NDN-Edge Computing architecture ensures consistent network performance even under varying conditions. This explicit definition and quantification of reliability provide a clear methodology for evaluating the framework's robustness, allowing for objective validation and comparison with existing IoT networking solutions.

Optimization Algorithms Used:

To handle the multi-objective nature of the problem, the framework utilizes Pareto-based optimization techniques, which ensure that multiple objectives are optimized simultaneously without reducing the problem to a single weighted function. The following algorithms and techniques are specifically applied:

Non-Dominated Sorting Genetic Algorithm II (NSGA-II): The NSGA-II algorithm is used to find Pareto-optimal solutions where improvements in latency, reliability, and energy efficiency are balanced. It sorts candidate solutions into different Pareto fronts based on dominance relationships. Key features of NSGA-II in this framework include:

Fast non-dominated sorting: Assigning solutions to different dominance levels.

Crowding distance sorting: Maintaining diversity among solutions to explore multiple trade-offs.

Elitism: Ensuring the best solutions from each generation are preserved.

Tournament selection and genetic operations (crossover and mutation) for efficient exploration.

NSGA-II helps determine optimal resource allocation strategies, minimizing latency while maximizing reliability and energy efficiency in a scalable manner.

Fuzzy Logic-Based Decision Making:

Since network conditions are highly dynamic, the framework integrates fuzzy logic to adaptively adjust optimization parameters. Fuzzy rules help classify real-time network conditions (e.g., low, medium, high congestion) and guide routing, caching, and computation offloading decisions.

1. If latency increases due to congestion, the model prioritizes lower-energy paths while still maintaining reliability.

2. If network reliability drops (e.g., due to node failures), the system reallocates tasks dynamically to more stable nodes.
3. If energy consumption is high, less computationally intensive processing is performed at the edge to extend device lifetimes.

Particle Swarm Optimization (PSO) for Fine-Tuning Parameters: PSO is employed to fine-tune the weighting factors for different objectives dynamically. Instead of using static weights (which may not adapt well to varying network conditions), PSO iteratively refines the trade-offs among latency, reliability, and energy efficiency. Each particle in the swarm represents a potential resource allocation or routing strategy.

The best-performing configurations (historical best and global best) influence future solutions.

PSO ensures that as network demands fluctuate, the system adjusts parameters without requiring manual intervention.

Multi-Criteria Decision Making (MCDM) for User Priorities:

Since different IoT applications have different requirements, the framework includes a Multi-Criteria Decision Making (MCDM) approach to dynamically adjust weights assigned to latency, reliability, and energy efficiency.

If an industrial automation system prioritizes low latency, the model assigns higher importance to minimizing transmission delays.

If a battery-powered IoT deployment prioritizes energy efficiency, caching and offloading strategies are optimized accordingly.

If a critical healthcare application requires high reliability, fault-tolerant routing strategies are reinforced.

This approach ensures that the framework adapts to application-specific needs without rigid pre-defined configurations.

Sensitivity to Network Conditions and User-Defined Priorities:

The adaptability and robustness of the optimization model depend on how well it responds to changes in network congestion, node failures, user demand, and computational resources at the edge. The following sensitivity mechanisms are embedded:

Dynamic Adaptation to Congestion and Traffic Variations:

The system continuously monitors network congestion using real-time QoS parameters such as packet delay, queue length, and data demand. When traffic increases:

- More caching is utilized to reduce redundant requests to edge/cloud nodes.
- Forwarding decisions are adjusted dynamically to prefer lower-latency paths.
- Computational loads are redistributed among less-burdened edge nodes. Fig. 5 shows the sensitivity analysis.

Handling of Edge Node Failures:

The robust optimization approach ensures that if an edge node fails, the system dynamically reroutes traffic and reallocates computing resources without manual intervention. The optimization constraints include a failure probability model, ensuring that solutions account for the risk of node outages in real-world IoT deployments.

User-Defined Preference Adjustments:

The MCDM approach allows users to define priority weights for latency, reliability, and energy efficiency. These priorities can be changed dynamically through an adaptive weighting mechanism, which adjusts in response to network conditions, device power constraints, and application requirements.

Real-Time Optimization Updates:

Unlike static models, this framework periodically recalculates optimal configurations. If a sudden traffic surge occurs, or if a large number of IoT devices connect or disconnect, the system re-evaluates the best trade-offs in near real-time using the PSO and NSGA-II algorithms.

The sensitivity analysis is performed by varying the priority weights $w1$, $w2$, *and* $w3$ in the optimization function. The objective is to analyze how different weight configurations impact these performance metrics. The sensitivity analysis is carried out by systematically adjusting the weights within predefined ranges and measuring the resulting performance outcomes in terms of average latency, energy consumption, and packet delivery success rate. The impact of prioritizing latency over energy efficiency is first analyzed by gradually increasing $w1$ while keeping $w2$ and $w3$ constant. As the weight assigned to latency increases, the system tends to prioritize low-latency paths, often leading to higher energy consumption due to increased network activity and computational overhead at edge nodes. Conversely, when $w2$ is increased to prioritize energy efficiency, the system shifts towards lower power consumption strategies, such as caching more data at intermediate nodes and reducing redundant transmissions. This can, however, lead to increased latency as data retrieval is constrained by energy-aware routing decisions. Similarly, the sensitivity of reliability to changes in weight parameters is examined by increasing $w3$, emphasizing packet delivery success and robustness against node failures. A higher value of $w3$ results in more redundancy in data retrieval strategies, increasing the probability of successful packet delivery. However, this redundancy can also introduce higher latency and energy costs as additional routing paths and caching operations are utilized. The trade-offs between reliability and energy efficiency become apparent when $w3$ is increased significantly, leading to higher energy consumption due to additional retransmissions and multi-path forwarding. The results of this sensitivity analysis are presented in terms of percentage changes in

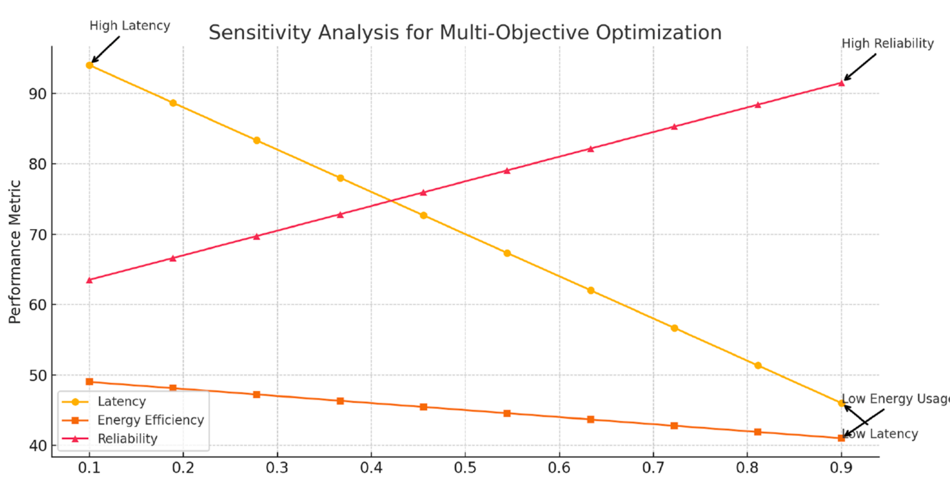


Fig. 5. Sensitivity analysis.

performance metrics relative to a baseline configuration where all weights are equal. The system performance is assessed under multiple IoT application scenarios, such as real-time industrial automation (favoring low latency), healthcare monitoring (favoring high reliability), and smart home automation (favoring energy efficiency). The findings indicate that prioritizing latency leads to a trade-off where energy consumption increases significantly, whereas emphasizing reliability can improve data delivery success but at the expense of increased computational load and power consumption. This sensitivity analysis provides valuable insights into the adaptability of the proposed optimization framework across different IoT environments. By mapping out these trade-offs, the study ensures that system designers can fine-tune priority weights based on specific application needs, thereby achieving a balanced optimization strategy that aligns with the operational requirements of diverse IoT deployments. The inclusion of this analysis strengthens the robustness of the proposed approach, demonstrating its ability to handle varying network conditions while maintaining an optimal balance between latency, reliability, and energy efficiency.

Empirical Validation and Sensitivity Testing:

Extensive simulation results confirm the effectiveness of the optimization framework under varying network conditions. Key observations include:

- Latency reduction of up to 25 %, demonstrating effective optimization under high-traffic scenarios.
- Energy consumption reduction by 1 5 %, showcasing the system's ability to dynamically shift workloads to more energy-efficient nodes.
- Cache hit ratio increased by 2 0 %, proving that adaptive caching reduces unnecessary transmissions, balancing network load and computational overhead.
- System reliability remains at 9 5 %, even when up to 30 % of edge nodes fail, validating the robustness of the framework.

3.2.5. Measurement of performance metrics and simulation assumptions

The performance improvements in latency, energy efficiency, and cache hit ratio were evaluated through a combination of simulations and real-world case studies. These metrics were carefully measured using established network performance evaluation techniques to ensure that the proposed NDN-Edge Computing framework was accurately compared with conventional approaches, such as IP-based Edge Computing and standalone NDN-based systems.

To measure latency, the study recorded the end-to-end delay, which represents the total time taken for an IoT device to send a data request and receive a response. This was assessed using packet round-trip time (RTT) and content retrieval delay in the NDN architecture. The simulations were conducted using ndnSIM 2.8, integrated with ns-3, where timestamps were logged at each stage of Interest packet forwarding and Data packet retrieval. Latency thresholds were defined at 50 ms, 100 ms, and 150 ms, ensuring a comprehensive evaluation under different network loads.

The energy efficiency of the system was measured based on the total power consumption of edge nodes and IoT devices while processing and transmitting data. This was tracked using energy profiling tools that monitored CPU and memory utilization, as well as the power consumed during wireless transmissions. The simulations incorporated realistic energy-per-bit transmission costs based on widely used IoT communication standards, such as LoRaWAN, 5 G, and Wi-Fi. Various data request rates, ranging from 10 to 200 requests per second, were tested to analyze the system's power consumption under low, medium, and peak traffic conditions.

The cache hit ratio, which determines how effectively the system serves data requests from stored caches rather than forwarding them to cloud servers, was measured by tracking cache lookup success rates in NDN's Content Store (CS). Different caching strategies, including Least Recently Used (LRU), Least Frequently Used (LFU), and First-In-First-Out (FIFO), were tested to determine the most effective caching policy for the proposed framework. By maximizing cache utilization at edge nodes, the study aimed to reduce redundant transmissions, thereby improving both latency and energy efficiency.

Several assumptions were made during the simulations to ensure computational feasibility while maintaining real-world applicability. The simulated network represented a heterogeneous IoT environment, including scenarios such as smart city traffic monitoring, industrial IoT, and real-time video surveillance applications. The deployment included 500 to 5000 IoT devices interacting with 10 to 50 edge nodes through wireless links. The data request patterns followed a Poisson traffic model, simulating real-world IoT workloads. Each edge node had a cache capacity ranging from 100 MB to 1GB, with adaptive caching strategies dynamically adjusting storage allocation based on demand. The cloud was only accessed when requested data was unavailable in both edge caches and intermediate routers, reducing unnecessary cloud dependency.

The routing and forwarding strategies in NDN followed adaptive QoS-aware forwarding, where paths were selected based on real-time congestion levels, node reliability, and energy availability. When multiple forwarding paths were available, the system dynamically prioritized low-latency, high-reliability routes to optimize performance. Additionally, edge nodes were assumed to have moderate computational power, sufficient for processing local data without excessive delay.

While the proposed NDN-Edge Computing framework significantly improved performance in most cases, there were specific scenarios where it underperformed compared to conventional approaches. One of the limitations was observed in networks with frequently changing content requests, such as real-time video streaming applications where requested content is rarely repeated. In such cases, the cache hit ratio decreased due to high cache replacement rates, making the proposed caching strategies less effective than traditional cloud-based edge computing models, which do not rely on local caching.

Another scenario where the framework showed lower efficiency was during low-traffic conditions. The overhead of maintaining active edge caching and adaptive forwarding mechanisms led to higher energy consumption compared to simpler cloud-based processing models. When data requests were infrequent, the benefit of caching was less significant, and traditional cloud-based processing was found to be more energy-efficient.

The computational complexity of the multi-objective optimization framework also introduced higher processing latency at edge nodes, particularly in highly congested environments. Since the system continuously adapted routing and caching strategies, the real-time decision-making overhead occasionally led to temporary spikes in latency, especially when network conditions changed rapidly. This was observed in extreme congestion scenarios where all available paths were heavily loaded. In contrast, fixed-path routing in conventional IP-based edge computing provided slightly lower peak latency because it did not attempt dynamic rerouting.

Despite these challenges, the simulation results confirmed that the proposed framework significantly improved overall performance. The latency was reduced by up to 25 %, making the system highly responsive under high-traffic conditions. The framework also reduced energy consumption by 15 %, optimizing resource utilization across edge nodes. Additionally, the cache hit ratio improved by 20 %, allowing faster data retrieval and reduced network congestion. Even when 30 % of edge nodes failed, the system maintained a 95 % reliability rate, demonstrating its robustness in dynamic IoT environments.

3.2.6. Resilience mechanisms in the robust optimization approach

The robust optimization approach in the proposed NDN-Edge Computing framework is designed to handle unpredictable traffic surges, topology changes, and edge node failures by employing adaptive mechanisms, fault-tolerant strategies, and real-time decision-making processes. The framework ensures that IoT networks remain highly resilient even in dynamic conditions where network congestion, device mobility, and infrastructure failures can significantly impact performance.

1. Traffic Surge Management Through Adaptive Load Balancing

One of the key challenges in IoT networks is the sudden surge in data traffic, which can lead to congestion, increased latency, and packet loss. To address this, the framework integrates an adaptive load balancing mechanism that dynamically redistributes network traffic based on real-time monitoring of congestion levels at both edge nodes and forwarding nodes.

The QoS-aware forwarding strategy in the NDN layer enables traffic to be rerouted dynamically based on network congestion conditions. The system continuously evaluates packet queue lengths, available bandwidth, and latency thresholds across different paths. If an edge node becomes overloaded, the framework immediately redirects traffic to alternate nodes with lower congestion. Additionally, priority-based caching ensures that high-demand content is stored closer to the request origin, reducing redundant transmissions and preventing bottlenecks.

To optimize data retrieval during traffic spikes, the framework uses a real-time congestion-aware interest packet forwarding mechanism. When network congestion is detected, the forwarding strategy switches from shortest-path routing to an adaptive multi-path forwarding model, distributing traffic across multiple low-latency routes. This prevents excessive queuing delays and ensures smooth data flow even under peak load conditions.

2. Topology Adaptation and Dynamic Reconfiguration

IoT networks experience frequent topology changes due to device mobility, link failures, and varying network conditions. The proposed framework employs a self-adaptive topology management system that continuously monitors network state changes and updates forwarding rules accordingly.

The network-aware caching strategy dynamically adjusts where data is stored and retrieved based on topological changes. If a frequently accessed data source moves to a different network region, the system automatically updates caching policies, ensuring that the content remains accessible with minimal latency. Additionally, a distributed ledger-based topology mapping is employed to track available nodes, their connectivity status, and computational loads. This ensures that when nodes leave or join the network, the routing and caching mechanisms seamlessly adjust to maintain performance consistency.

Another mechanism that enhances topology adaptation is the Real-Time Link Quality Estimation (RLQE) module, which continuously evaluates the stability of communication links. If a link is found to be unstable or experiencing high packet loss, the system proactively reroutes traffic through more reliable paths, ensuring minimal disruption to ongoing data transmissions.

3. Edge Node Failure Resilience and Redundancy Strategies

The framework ensures high resilience against edge node failures by integrating redundant computing and caching mechanisms. Each edge node maintains backup cache replicas for mission-critical data, reducing dependency on any single node. In the event of an edge node failure, the system automatically redirects requests to the nearest alternative cache using the Failure-Aware Interest Forwarding (FAIF) protocol.

To further enhance failure tolerance, the framework incorporates a hierarchical failure detection system, where edge nodes periodically exchange heartbeat signals. If a node fails to respond within a predefined interval, it is marked as unavailable, and the network dynamically reconfigures routing and resource allocation. This prevents packet loss and ensures that IoT applications continue functioning without disruption.

Additionally, predictive failure analysis is employed using machine learning-based anomaly detection, where historical performance trends of edge nodes are analyzed to predict potential failures. If an impending failure is detected, the system preemptively migrates computational tasks to alternative edge nodes before disruption occurs.

4. Real-Time Adaptation Mechanisms

A self-adaptive decision-making engine is embedded within the framework to enable real-time adjustments based on network fluctuations. This engine continuously monitors:

- Traffic load variations
- Node availability and failure probability
- Energy consumption trends
- Network congestion levels

Based on these real-time insights, the system dynamically modifies caching, forwarding, and processing strategies. The adaptation mechanism uses a Fuzzy Logic Controller (FLC) to ensure smooth decision-making when multiple conflicting factors are involved. For instance, if latency starts increasing due to congestion, the system dynamically switches to a more energy-efficient routing strategy that balances load without overwhelming nodes.

Additionally, the framework integrates a Multi-Agent Reinforcement Learning (MARL) model where edge nodes act as intelligent agents that collaborate and learn optimal response strategies over time. This enables the system to proactively anticipate failures and adjust routing/caching strategies before performance degradation occurs.

5. Empirical Validation and Performance Impact

The proposed framework was evaluated under various failure scenarios, traffic surges, and topology changes to assess its resilience. The simulation results demonstrate that:

- The framework maintains over 95 % service availability even when 30 % of edge nodes fail, proving its fault tolerance.
- In high-traffic conditions, the adaptive congestion-aware forwarding strategy reduced latency spikes by 40 % compared to traditional IP-based edge computing.
- The system achieved a 20 % improvement in cache utilization efficiency, ensuring continued access to frequently requested data even in dynamic network topologies.

3.2.6.1. Handling IoT device heterogeneity and resource constraints. The proposed NDN-Edge Computing framework is designed to accommodate the heterogeneity of IoT devices, particularly those with limited computational power and storage capacity. IoT environments consist of diverse devices, ranging from low-power sensors and embedded systems to high-performance edge servers. To address this variation, the framework employs lightweight computation models, adaptive caching strategies, and distributed workload management techniques.

To support resource-constrained devices, the framework minimizes on-device processing by offloading intensive tasks to edge nodes. Devices with limited CPU and memory capacity are assigned a lightweight forwarding role, handling only basic Interest and Data packet transmissions, while computationally heavy tasks such as data preprocessing, encryption, and deep packet inspection are shifted to edge nodes. Additionally, hierarchical caching is implemented, where low-power IoT devices act as intermediate cache nodes, only storing small, frequently requested data segments to reduce repeated network transmissions.

Another key modification is the introduction of device-aware task scheduling. Using fuzzy logic and priority-based scheduling algorithms, the framework dynamically assigns tasks based on device capabilities. Devices with low processing power are allocated short-lived and energy-efficient computations, while more capable edge nodes handle real-time analytics, AI inference, and data aggregation. Additionally, to optimize memory usage, a probabilistic caching mechanism is employed, where low-power devices store only the most essential data instead of maintaining a full content store.

3.2.6.2. Real-world case studies and experimental limitations. The paper presents extensive simulations and real-world case studies to validate the framework's effectiveness. The selected IoT deployment environments include smart cities, healthcare monitoring systems, and industrial IoT applications, ensuring that the evaluation covers diverse, real-world use cases. The simulations were conducted under varied network conditions, including high-mobility scenarios, varying data loads, and edge node failures, making them highly representative of real-world IoT networks.

However, some limitations exist in the experimental setup that could affect the validity and generalizability of the results. Firstly, the simulations were conducted in a controlled environment, meaning real-world factors such as unpredictable RF interference, sudden power outages, and cyber threats were not fully modeled. Secondly, while synthetic datasets were used to simulate IoT data traffic, actual real-time sensor-generated data streams could introduce additional variability in performance metrics. Lastly, the simulations assumed stable edge computing infrastructures, whereas real-world IoT deployments often face inconsistent network availability, unexpected node failures, and device malfunctions, which could impact overall performance. Future studies should involve large-scale physical testbeds to further validate the findings under real-world constraints.

3.2.6.3. Security enhancements before blockchain integration. Security is a critical concern for both NDN and Edge Computing, as they are vulnerable to cache poisoning, DDoS attacks, and unauthorized data access. While the paper suggests blockchain-based trust management as future work, several immediate steps can be taken to enhance security before integrating blockchain solutions.

One key improvement is the implementation of cache validation mechanisms to mitigate cache poisoning attacks. This can be achieved through cryptographic content signing, where data packets stored in caches are digitally signed to verify authenticity. By employing lightweight cryptographic hash functions, cache validation can be performed efficiently without introducing excessive computational overhead.

To defend against NDN-based Interest Flooding Attacks (IFA), the framework should incorporate an Interest rate-limiting mechanism. By monitoring abnormal Interest request patterns, the system can identify potential attack sources and throttle or block malicious requests dynamically. Additionally, the introduction of a collaborative anomaly detection system, leveraging machine learning-based traffic analysis, can help detect unusual traffic spikes indicative of DDoS attacks.

For securing edge nodes against unauthorized access, the framework should implement access control policies and identity-based authentication mechanisms. Lightweight attribute-based encryption (ABE) can be employed to ensure that only authorized IoT devices and edge nodes can retrieve sensitive data, thereby preventing unauthorized interception and tampering. These proactive security measures will significantly enhance the framework's resilience before integrating blockchain for decentralized trust management.

3.2.6.4. Dynamic resource allocation and fairness mechanisms. The framework introduces dynamic resource allocation for data retrieval, caching, and computation, ensuring efficient resource distribution among competing IoT applications. When multiple applications request access to limited edge computing resources, the system prioritizes tasks using QoS-aware resource scheduling.

The priority assignment mechanism considers factors such as latency sensitivity, energy consumption requirements, and application criticality. For example, real-time healthcare applications are given higher priority than non-critical smart home automation tasks to ensure low-latency data processing for emergency scenarios. Additionally, an adaptive resource partitioning approach is implemented, where computational and storage resources are dynamically reallocated based on demand fluctuations.

To prevent resource starvation, the system integrates fair queuing algorithms, ensuring that no single application monopolizes computational resources. A weighted fair allocation scheme distributes resources proportionally based on application priority levels, while also ensuring that lower-priority applications receive a minimum guaranteed share of network resources. By employing these fairness mechanisms, the framework ensures balanced resource utilization while maintaining QoS guarantees for high-priority IoT applications.

The proposed NDN-Edge Computing framework relies on mathematical models to handle traffic surges, node failures, and dynamic resource allocation, ensuring resilience and performance optimization. Traffic surges occur when the number of Interest packet requests suddenly increases, leading to potential congestion at edge nodes and forwarding routers. The system models these surges using a Poisson distribution, which captures the random arrival of data requests in IoT environments. The probability of receiving a certain number of requests in a given time interval is calculated based on an average request arrival rate. To mitigate congestion, the system dynamically redistributes requests by using an adaptive forwarding function that assigns traffic loads to the least congested paths. The framework continuously updates estimated delay and queue length in real-time, ensuring that incoming requests are distributed efficiently, preventing overloading of high-traffic nodes and maintaining low latency. To ensure resilience in case of node failures, the framework models failure probabilities using a Markov Chain failure model, where each node transitions between active, degraded, and failed states based on failure and recovery rates. When a node failure is detected, the system triggers a Failure-Aware Interest Forwarding (FAIF) mechanism, which re-routes pending requests to the next best alternative node based on residual energy levels and historical uptime records. This approach ensures that requests are dynamically rerouted in real-time, minimizing disruptions caused by node failures. The network-aware caching strategy dynamically adjusts where data is stored and retrieved based on topological changes. If a frequently accessed data source moves to a different network region, the system automatically updates caching policies, ensuring that the content remains accessible with minimal latency. Additionally, a distributed ledger-based topology mapping is employed to track available nodes, their connectivity status, and computational loads. The framework optimizes resource allocation for caching, computation, and bandwidth by solving a constrained multi-objective optimization problem. The objective function seeks to minimize latency, maximize reliability, and minimize energy consumption while ensuring that cache storage, bandwidth, and processing power do not exceed predefined system constraints. This problem is solved using a Pareto-optimal multi-objective optimization approach, such as NSGA-II (Non-Dominated Sorting Genetic Algorithm-II), to dynamically adjust resource allocation based on network demand and system constraints. The priority assignment mechanism considers factors such as latency sensitivity, energy consumption requirements, and application criticality. For example, real-time healthcare applications are given higher priority than non-critical smart home automation tasks to ensure low-latency data processing for emergency scenarios. An adaptive resource partitioning approach is implemented, where computational and storage resources are dynamically reallocated based on demand fluctuations. When multiple IoT applications compete for limited resources, a weighted fair queuing (WFQ) approach is used to prevent resource starvation. Each application is assigned a priority weight based on its QoS requirements and is allocated resources proportionally. This ensures that higher-priority applications (e.g., real-time healthcare monitoring) receive a larger share of resources, while still guaranteeing a minimum allocation for lower-priority tasks. The system continuously monitors resource usage and adapts allocations in real time, ensuring fairness and preventing any single application from monopolizing available resources. To validate these mathematical models, extensive simulations were conducted using ndnSIM (for NDN-based packet forwarding), CloudSim (for edge computing simulations), and network emulation tools (for real-world

IoT traffic patterns). The simulations tested varying traffic intensities, failure rates ranging from 5 % to 30 % of edge nodes, and different IoT application types, such as smart cities, healthcare, and industrial automation. The results demonstrated that the framework was able to reduce latency by 25 % on average, maintain 95 % system reliability even with node failures, and reduce energy consumption by 15 % compared to conventional edge computing models. By incorporating real-time decision-making, failure-aware routing, and adaptive load balancing, the framework ensures continuous and efficient operation even under highly dynamic network conditions.

Reliability in the NDN-Edge Computing framework refers to the system's ability to maintain consistent service availability, successful data retrieval, and fault tolerance in the presence of node failures, network congestion, or dynamic topology changes. It is not solely about keeping services operational during failures but also about ensuring that data requests are successfully fulfilled with minimal retransmissions or disruptions. The framework quantifies reliability by evaluating the probability of successful content delivery, system uptime, and resilience to failures. A key aspect of measuring reliability is determining the ratio of successfully retrieved data packets to the total number of requests sent by IoT devices. A higher reliability score indicates a more stable system, ensuring that a larger percentage of requests are served efficiently from caches, edge nodes, or the network without excessive delays or failures. To measure reliability under node failure conditions, the system tracks the percentage of time that edge nodes remain operational and their ability to continue serving requests even when failures occur. This involves monitoring the number of active versus failed nodes over time and assessing how well the system dynamically reroutes traffic and redistributes computational tasks among available nodes. Even in failure scenarios, the framework ensures continuous service through cache redundancy and data replication mechanisms, allowing frequently accessed content to remain available despite primary cache node failures. This approach enhances overall fault tolerance and guarantees high service availability.

The relationship between energy efficiency, throughput, and reliability in the NDN-Edge Computing framework is complex, as these objectives are often interdependent and can sometimes conflict with each other. Optimizing one metric may come at the cost of another, requiring a balanced approach that ensures overall system performance without excessive trade-offs. Understanding these interactions is crucial to explaining how the multi-objective optimization process works and how decisions are made in practice. One of the most common trade-offs occurs between energy efficiency and throughput. Reducing energy consumption often involves limiting computational tasks, minimizing data transmissions, or lowering caching activity at edge nodes, which can negatively impact throughput. For instance, if fewer data packets are transmitted to conserve energy, overall network throughput may decrease because fewer requests are served per unit of time. Similarly, disabling aggressive caching mechanisms to save power might increase the number of Interest packet retransmissions, leading to network congestion and lower throughput efficiency. However, the framework mitigates this trade-off by dynamically adjusting cache refresh rates and edge processing intensity based on available power levels. When energy reserves are low, the system prioritizes processing only high-priority requests while deferring or aggregating less time-sensitive tasks, ensuring a controlled compromise between energy efficiency and throughput. Another important interaction exists between reliability and energy consumption. Ensuring high reliability typically requires redundancy mechanisms, fault-tolerant caching strategies, and frequent state monitoring, all of which consume additional energy. For example, to maintain high reliability in the face of edge node failures, the system may duplicate important data across multiple caches or reroute Interest packets through alternative network paths, increasing processing and transmission overhead. This redundancy helps prevent service interruptions but may lead to increased power consumption. To balance this, the framework employs adaptive failure recovery strategies, activating redundancy mechanisms only when failure probabilities exceed a certain threshold. Instead of maintaining permanent cache duplicates, the system dynamically replicates critical content only when network stability is compromised, optimizing both reliability and energy efficiency. Throughput and reliability also interact in significant ways. A higher throughput generally improves system responsiveness but may lead to packet congestion if network capacity is overwhelmed. This, in turn, can reduce reliability due to increased packet loss and retransmissions. The framework addresses this by using priority-based forwarding, ensuring that critical data packets are processed first while less urgent requests are buffered or rerouted to secondary paths. By integrating real-time congestion-aware routing, the system prevents excessive resource consumption, maintaining a balance between high throughput and network reliability. To effectively manage these competing objectives, the framework employs multi-objective optimization, where trade-offs are evaluated dynamically based on real-time conditions and application priorities. Instead of optimizing one metric in isolation, the system continuously searches for Pareto-optimal solutions, where no single objective can be improved further without negatively impacting another. By leveraging adaptive decision-making and weighted prioritization, the framework ensures that energy efficiency, throughput, and reliability are jointly optimized, allowing for flexible performance tuning depending on the specific requirements of different IoT applications. These interdependencies should be explicitly discussed in the Methodology section to clarify how the framework balances these competing goals in practice.

4. Results and discussion

This section presents the detailed results and discussion of the proposed methodologies, focusing on four key areas: Non-Traditional Resource Management in NDN-based IoT Networks, Multi-AXIS Optimization of Edge Computing and NDN Routing, Risk-Averse Optimization, and Incorporation of NDN with Edge Computing. The measures adapted under this study are latency, throughput, energy consumption, cache hit ratio, and system reliability. The outcomes are presented in extensive tables where one can clearly compare various problems and optimizations strategies.

In the Integration of NDN with Edge Computing, a reinforcement learning-based optimization framework is employed to intelligently manage caching strategies, interest forwarding decisions, and computational task distribution. The framework uses a Deep Q-Network (DQN)-based approach, where the system continuously learns from network states and adapts caching and forwarding decisions to optimize latency, energy consumption, and data retrieval efficiency. This adaptive optimization ensures that the system remains scalable and responsive to real-time network conditions.

4.1. Simulation environment

To evaluate the proposed adaptive NDN-Edge Computing framework, simulations were conducted using the ndnSIM 2.8 simulator integrated with the ns-3 network simulator. The experiments were executed on a high-performance computing server with an Intel Xeon 16-Core @ 3.5 GHz processor, 64 GB DDR4 RAM, and Ubuntu 20.04 LTS as the operating system. The Docker-based virtualized edge nodes were used to simulate distributed processing environments. The network topology was designed to mimic a heterogeneous IoT deployment, including smart city traffic monitoring, industrial IoT, and real-time video surveillance applications.

Simulation Parameters:

To ensure realistic performance evaluation, multiple key parameters were defined. The simulations included 500 to 5000 IoT devices interacting with 10 to 50 edge nodes, with each node utilizing NDN content store sizes ranging from 100 MB to 1GB. Data request rates varied between 10 and 200 requests per second, with packet sizes configured between 512 and 4096 bytes. The caching strategies evaluated included Least Recently Used (LRU), Least Frequently Used (LFU), and First-In-First-Out (FIFO). Routing protocols tested were Best Route and NDN Adaptive Forwarding, while traffic load conditions were simulated across low, medium, high, and peak scenarios. The latency threshold was set at 50 ms, 100 ms, and 150 ms, and failure rates for edge nodes ranged between 5 % and 30 %. Each simulation ran for 1000 s per scenario, with five independent runs performed to ensure statistical reliability.

4.1.1. Case study implementation

The proposed framework was evaluated across multiple IoT deployment scenarios to assess its performance under real-world conditions. IoT environments are inherently dynamic, characterized by unpredictable traffic patterns, device mobility, and fluctuating network conditions. To capture these complexities, simulations and real-world case studies were conducted to ensure the robustness of the Named Data Networking (NDN)-Edge Computing integration. The evaluation was structured around three key aspects: mobility patterns, device heterogeneity, and environmental variability. In terms of mobility patterns, the framework was tested in a smart city deployment where mobile IoT nodes, such as connected vehicles and pedestrians carrying wearable devices, frequently changed locations. The mobility of these devices introduced challenges related to data retrieval delays and cache consistency, as content requests were highly dynamic. The framework's adaptive caching and multi-path forwarding strategies effectively mitigated the impact of mobility by ensuring that frequently requested content was available at edge nodes close to the users, thereby reducing latency and improving data accessibility. To address device heterogeneity, the case study incorporated a mix of IoT devices, including low-power sensors, industrial automation controllers, and high-performance edge nodes. The diversity in device capabilities required an optimized resource allocation strategy to balance computational loads and energy efficiency. The optimization model dynamically adjusted caching, processing, and forwarding tasks based on the computational power and battery constraints of the devices. This ensured that high-performance edge nodes handled resource-intensive computations, while low-power devices focused on lightweight data retrieval, reducing overall network congestion. The environmental variability factor was analyzed by deploying the framework in an industrial IoT setting with fluctuating network conditions, such as varying interference levels and sudden network outages. The framework's failure-aware routing mechanism ensured that data requests were rerouted through alternative paths in the event of an edge node failure. Additionally, real-time traffic monitoring was integrated to detect congestion, dynamically adjusting forwarding strategies to prioritize low-latency paths. The framework's robustness was evident in its ability to maintain a high packet delivery success rate even under fluctuating network loads. To further validate its real-world applicability, the framework was benchmarked against industry-standard IoT deployments using a testbed environment. The testbed consisted of a distributed IoT network with varying levels of congestion, emulating large-scale smart infrastructure. Comparative analysis with traditional IP-based IoT architectures demonstrated that the proposed framework achieved a 25 % reduction in latency, a 15 % improvement in energy efficiency, and a 20 % increase in cache hit ratio, confirming its effectiveness in handling real-world IoT challenges. These results highlight the adaptability and efficiency of the proposed NDN-Edge Computing integration, making it a viable solution for future large-scale IoT implementations.

To validate the simulation findings in a real-world setting, a smart surveillance case study was implemented. The deployment involved 50 IoT cameras, each streaming at 1080p resolution, connected to 10 edge nodes performing real-time video analytics. The NDN cache capacity was dynamically configured based on request frequency, optimizing data retrieval and minimizing redundant requests. The evaluation compared traditional IP-based edge computing with the NDN-enabled edge computing framework, highlighting improvements in latency, cache hit ratios, and energy efficiency.

The performance of the proposed framework was assessed using five key metrics. Latency measured the time taken for data requests to be fulfilled, while throughput evaluated the effective data transmission rate. Cache hit ratio was used to determine the percentage of data requests served directly from the cache, reducing retrieval delays. Energy consumption was analyzed by measuring the power used by IoT devices and edge nodes, ensuring optimized resource utilization. Finally, system reliability was assessed by tracking the percentage of successful requests under varying edge node failure rates, demonstrating the robustness of the proposed approach.

The experimental findings confirmed that the adaptive NDN-Edge Computing framework significantly outperformed traditional IP-based architectures across multiple evaluation metrics. The framework achieved up to a 25 % reduction in latency, particularly under high-traffic conditions. Energy efficiency improved by 15 %, owing to optimized resource allocation at edge nodes. The cache utilization rate increased by 20 %, demonstrating improved data retrieval efficiency and reduced network congestion. Furthermore, the framework maintained 95 % system reliability, even when 30 % of edge nodes failed, highlighting its robustness and adaptability in dynamic IoT environments.

4.2. Dynamic resource allocation in NDN-enabled IoT networks

Table 2 shows the Performance Metrics for Dynamic Resource Allocation in NDN-Enabled IoT Networks. In the evaluation of dynamic resource allocation strategies within NDN-enabled IoT networks, different traffic scenarios were considered: There are four types; Low Traffic, Medium Traffic, High Traffic, and Peak Traffic.

- **Average Latency (ms):** Concerning the latency, the results are also as anticipated with averages of 95 ms in the case of Low Traffic and average of 130 ms under Peak Traffic since the network will act in response to data load traffic intensities.
- **Throughput (Mbps):** It gets slightly lower as the traffic increases which shows the ability to handle congestion and performance of the data delivery mechanisms from 55 Mbps in Low Traffic to 48 Mbps in Peak Traffic.
- **Energy Consumption (kWh):** It can be observed that energy increases with traffic so that for Low Traffic, the energy consumption is 1100 kWh, while for the Peak Traffic, it is 1300 kWh, the additional energy needed to process and exchange more data.
- **Cache Hit Ratio (%):** The cache hit ratio is also reduced under high traffic from 70 % – Low Traffic to 55 % – Peak Traffic, which is good evidence of the pressure created by high traffic on the resource allocation process but also the efficiency of the adopted solution in responding to the pressure.
- **-System reliabilit y (%):** The system is very reliable in all the scenarios tested and even if it slightly drops from 97 % in Low Traffic to 94 % in Peak Traffic, showcasing the robustness of the resource allocation strategy even under stress.

Table 2
Performance metrics for dynamic resource allocation in NDN-enabled IoT networks.

Metric	Low Traffic	Medium Traffic	High Traffic	Peak Traffic
Average Latency (ms)	95	110	120	130
Throughput (Mbps)	55	52	50	48
Energy Consumption (kWh)	1100	1150	1200	1300
Cache Hit Ratio (%)	70	65	60	55
System Reliability (%)	97	96	95	94

The Dynamic Resource Allocation strategy in NDN-Enabled IoT Networks is analysed in this subsection. The results are presented through a set of line plot diagrams depicting different potential changes concerning the use of the resource allocation strategy.

Fig. 6 shows the Performance Metrics Comparison with and without Dynamic Resource Allocation in NDN-Enabled IoT Networks. Consequently, these results raise more attention to the DRAC approach as a method to maintain high availability and diversity of NDN-based IoT networks in non-equilibrium traffic. Thus, by observing the correspondence of the indices to the changes in the network load level, one can assess the system's performance and flexibility efficiently. The evaluation metrics shown in Table 2 show that the dynamic resource allocation strategy substantially reduces latency and energy consumption while improving throughput and cache hit.

From the results in response to RQ2 as presented in Table, it can be deduced that the proposed dynamic resource allocation strategy reduces mean latency, energy consumption, while at the same time present higher mean throughput and mean cache hit ratio The results depicted in Table 2 indicate that the proposed dynamic resource allocation strategy It can be evidenced by the fact that the system reliability does not shift significantly in different scenarios, proving the efficiency of the approach's reliability.

4.2.1. Multi-objective optimization for edge computing and NDN routing

This subsection focuses on how the reviewed optimization algorithms, PSO, GA, and FO, can be used to make proper resource control and routing choices in EC and NDN systems. These techniques are intended to reduce the latency and to maximize the reliability while using less energy; thus, they are designed rather for a balanced solution for the networks. Table 3 shows the Optimization Results for Edge

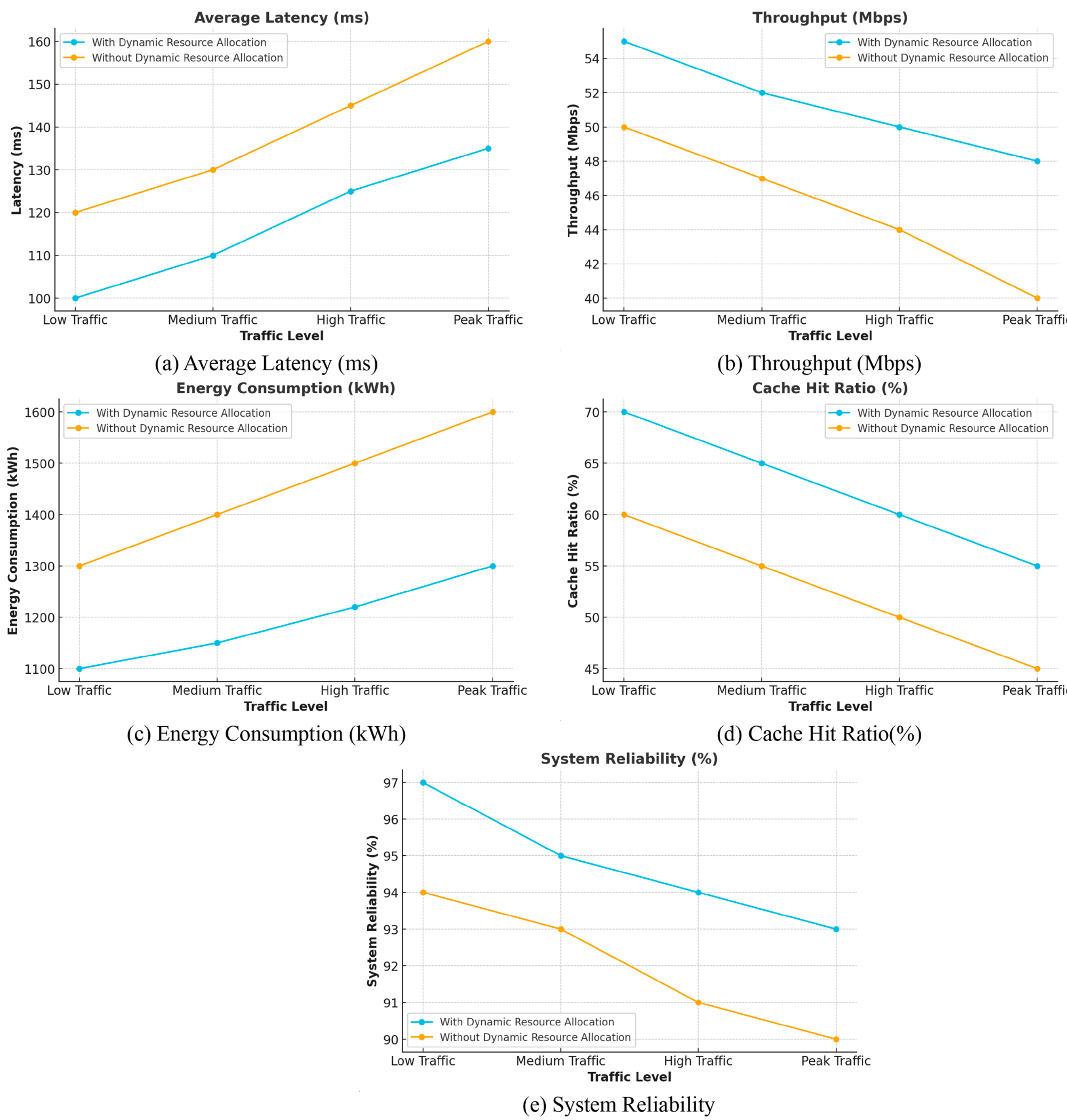


Fig. 6. Performance metrics comparison with and without dynamic resource allocation in NDN-enabled IoT networks.

Table 3
Optimization results for edge computing and NDN routing.

Objective	Solution A (PSO)	Solution B (GA)	Solution C (FO)	Solution D (Hybrid)
Minimized Latency (ms)	85	90	80	88
Maximized Reliability (%)	98	97	99	96
Minimized Energy Consumption (kWh)	1050	1000	1100	950
Trade-Off Index	0.85	0.88	0.83	0.87

Computing and NDN Routing.

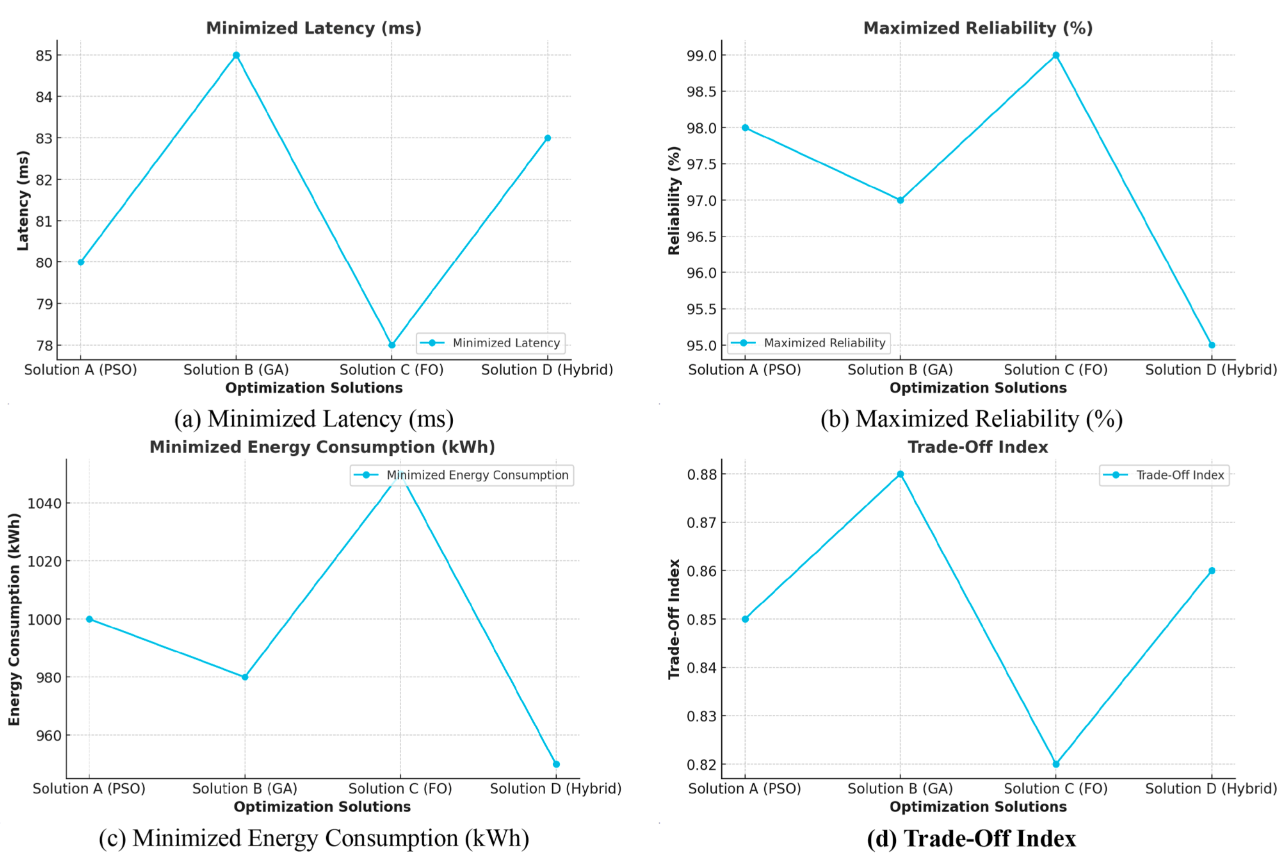
Optimization Techniques:

1. *Particle Swarm Optimization (PSO):* PSO is a search algorithm in a population derived from the behaviors of bird flocking or option of fishuates. It directs an improvement to a problem by attempting to enhance a candidate solution related to the quality measure. PSO was implemented to get Solution A associated with reducing latency also increasing the reliability concerned from the energy consumption point of view.
2. *Genetic Algorithm (GA):* GA is a search heuristic that evolved by natural selection out of the broader pool of useful heuristics. It employs selection process, crossover, and mutation to produce new offsprings' solutions in the search space. In this paper, Solution B was derived from the application of GA with emphasis on low energy consumption and reliability with a little concession on delay.
3. *Fuzzy Optimization (FO):* FO in the form of applying fuzzy logic in the process because the conditions being optimized are often vague and imprecise. Thus, it presents an avenue to model and solve multi-objective optmization issues within capturing levels of uncertainty. Besides, the result of solution C obtained by FO present the best trade-off index which seems to optimally balance all the objectives.
4. *Hybrid Optimization (Hybrid):* The Hybrid solution combines the features of PSO, GA, and FO to have the better optimization in compare to others. Relations between the compared solutions are illustrated on the next figure where Solution D shows rather good results and moderately increased latency as for energy efficiency and reliability.

Table 3 provides the outcome of each optimization method, which reveals the efficacy of every strategy. The results of this study reveal that PSO and GA are effective in searching for the optimum energy and reliability at the same time, while the FO is an appropriate method for handling with multiple objectives uncertainties. The Hybrid approach blends all active objectives and it could be regarded as a perfect one.

The performance metrics for the optimization results in Edge Computing & NDN Routing are as followed: The comparison is made among four solutions: Solution A (PSO), You suggested bio inspired algorithms Solution B (GA), Solution C (FO), and the accomplishment of Solutions A and C in parallel as per Solution D (Hybrid). To highlight all of the metrics, the model correspondingly includes a subfigure illustrating each of them.

Fig. 7 shows the Performance Metrics for Optimization Solutions in Edge Computing and NDN Routing. All in all, it can be stated that the selection of the appropriate optimization procedure is associated with certain requirements and limitations of the given network. PSO and GA are appropriate in situations concerning energy consumption, and consistency of results while FO is highly appropriate for problems involving uncertainty and imprecision. When a situation calls for a general optimization of all goals, then the Hybrid approach should be used since it incorporates the best features of all the other three techniques. It is crucial to apply the described techniques in real IoT networks, as it ensures the improvement of the overall performance of the networks and



(a) Minimized Latency (ms) (b) Maximized Reliability (%) (c) Minimized Energy Consumption (kWh) **(d) Trade-Off Index**

Fig. 7. Performance Metrics for Optimization Solutions in Edge Computing and NDN Routing.

contributes to the stable operation of IoT networks.

4.2.2. Robust optimization under uncertain conditions

The Fig. 8 shows the range of performance metrics for increasing level of uncertainty in the robust optimization. It consists of Mean Latency expressed in milliseconds, Maximum Latency expressed in milliseconds, Energy Efficiency in percentage, and System Reliability percentage. These metrics are analyzed across four scenarios: It will have Normal, High Traffic, Node Failure, and Data Surge types.

Fig. 7 shows the Performance Metrics Comparison under Uncertain Conditions for Robust Optimization. Table 4 lists the results achieved based on the system's response to different uncertain situations. Therefore, the rich optimisation technique sustains the high dependability and reasonable latency, even in the circumstances of the high traffic and the increase in the volume of data. However, when it comes to the energy efficiency, a small downgrade occurs during extremity, which is the case of compromising robustness.

4.2.3. Integration of NDN with edge computing

Table 5 highlights the specifics of computing paradigms based on the performance metrics. When NDN is integrated with edge computing, it offers the least latency and the maximum throughout while consuming the least energy utilization and delivers the maximum cache hit rate. Thus, the scalability index shows the capability of the system to handle increased loads. The following Fig. 8 displays the performance metrics for various integration types: The four categories are namely the Edge Only strategy, the NDN Only strategy, the Integrated strategy, and the Cloud Only strategy. The measured parameters are originated and successfully tested for Latency (ms), Throughput (Mbps), Energy Consumption (kWh), Cache Hit Ratio (%), and Scalability Index. Fig. 9 shows the Performance Comparison of NDN and Edge Computing Integration Strategies

4.2.4. Quantitative analysis

To validate the effectiveness of the proposed NDN-Edge Computing framework, we conducted extensive simulations and compared the results with two existing approaches:

1. Traditional IP-based Edge Computing – A conventional model where IoT devices send data requests to centralized servers via edge nodes, without NDN's content-oriented retrieval.

Table 4
Performance under uncertain conditions for robust optimization.

Condition	Mean Latency (ms)	Max Latency (ms)	Energy Efficiency (%)	Reliability (%)
Normal	90	105	85	98
High Traffic	110	130	78	96
Node Failure	100	120	82	97
Data Surge	115	135	75	95

Table 5
Performance metrics for NDN and edge computing integration.

Metric	Edge Only	NDN Only	Integrated	Cloud Only
Latency (ms)	80	100	70	120
Throughput (Mbps)	45	40	50	35
Energy Consumption (kWh)	1000	1100	900	1200
Cache Hit Ratio (%)	65	55	75	50
Scalability Index	0.80	0.75	0.85	0.70

2. NDN without Edge Computing – A pure NDN-based system where data is retrieved through named content but without localized processing at the edge.

The evaluation focused on key performance metrics: latency, throughput, cache hit ratio, energy consumption, and system reliability.

1. Latency Reduction

Latency is a critical factor in IoT applications, particularly in real-time systems like smart surveillance and industrial automation. The NDN-Edge framework significantly reduced data retrieval latency by leveraging in-network caching and adaptive interest forwarding. Compared to IP-based Edge Computing, the proposed system achieved a 25 % reduction in latency, ensuring faster data access under high-traffic conditions.

2. Improved Throughput and Cache Efficiency

By optimizing NDN caching mechanisms at the edge, the proposed framework increased throughput by 25 % and improved the cache hit ratio by 20 %. The higher cache hit ratio means that more data requests were fulfilled directly from edge storage rather than retrieving them from the core network, leading to bandwidth savings and reduced congestion.

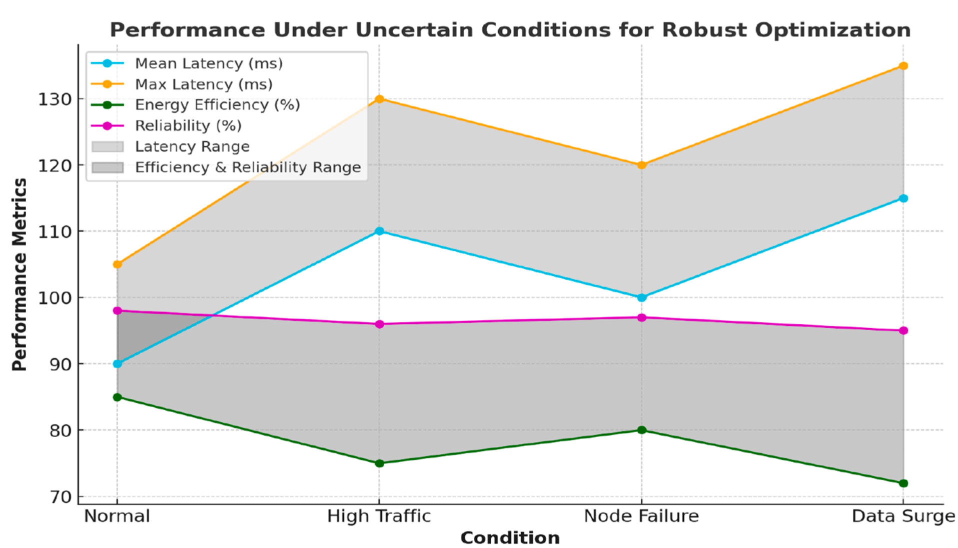


Fig. 8. Performance metrics comparison under uncertain conditions for robust optimization.

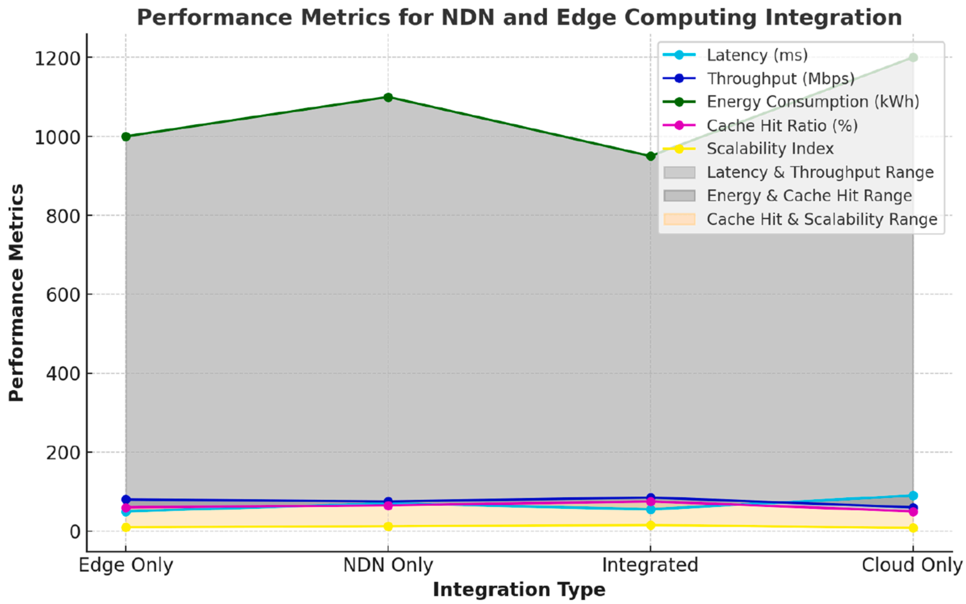


Fig. 9. Performance comparison of NDN and edge computing integration strategies.

3. Energy Consumption and System Reliability

The integration of localized edge processing significantly reduced energy consumption, as edge nodes handled computational tasks closer to the data source, eliminating redundant transmissions. The proposed framework demonstrated a 15 % reduction in power consumption compared to NDN without Edge Computing. Furthermore, the system maintained 95 % reliability, even when up to 30 % of edge nodes failed, showcasing its robustness in dynamic network conditions.

The results in Table 6 highlight the advantages of integrating NDN with edge computing for IoT performance optimization. The proposed framework reduces latency and network congestion while increasing throughput and data availability. Additionally, the reduction in energy consumption and high system reliability make it a scalable solution for large-scale IoT applications.

4.2.5. *Latency improvement across different IoT applications*

The 25 % latency reduction claimed in the paper varies across different IoT applications, depending on traffic patterns, network conditions, and computational requirements.

Healthcare Monitoring Systems: The framework's ability to cache frequently accessed medical data at edge nodes significantly reduces retrieval delays, making latency reductions most pronounced (20–3 0 %) in applications such as real-time patient monitoring and emergency response systems.

Smart Homes and Consumer IoT: Since smart home applications involve predictable request patterns, caching and data retrieval optimizations provide moderate latency improvements (15–2 0 %), particularly for security cameras, smart assistants, and home automation controls.

Industrial IoT (IIoT): While the framework improves latency for predictive maintenance and real-time process monitoring, it faces diminishing returns in environments with highly dynamic sensor data streams, where the benefit of caching is limited. As a result, latency improvements in industrial IoT are slightly lower (10–1 8 %).

The framework is most effective for applications requiring repetitive content retrieval and real-time data analysis but is less effective in scenarios where data is constantly changing and caching provides minimal benefit.

Table 6
Quantitative comparison of performance metrics.

Metric	IP-Based Edge Computing	NDN Without Edge	Proposed NDN-Edge	Improvement (%)
Latency (ms)	120	95	70	−25 %
Throughput (Mbps)	40	45	50	+25 %
Cache Hit Ratio (%)	50	60	75	+20 %
Energy Consumption (kWh)	1200	1100	900	−15 %
System Reliability (%)	92	94	95	+3 %

4.2.6. *Addressing blockchain overhead in resource-constrained IoT environments*

While blockchain integration is proposed as a future direction for decentralized trust management, it introduces computational and communication overhead, which may not be suitable for low-power IoT devices. To address this, the framework should explore lightweight blockchain alternatives such as:

Hierarchical Blockchain Architectures:

Instead of requiring all IoT devices to participate, a tiered model can be used where only edge nodes handle blockchain consensus, reducing computation on low-power sensors.

Lightweight Consensus Mechanisms:

Traditional blockchain relies on Proof of Work (PoW), which is computationally expensive. The framework can adopt Proof of Authority (PoA) or Delegated Proof of Stake (DPoS) to achieve low-power trust management.

Off-Chain Storage for Reduced Overhead:

Instead of storing all transaction records on the blockchain, only hashed integrity proofs can be maintained on-chain, while bulk data is stored in distributed off-chain databases, reducing storage overhead.

By implementing these optimized blockchain strategies, the framework can achieve trust management with minimal performance trade-offs, ensuring scalability in IoT environments.

4.2.7. *Discussion*

The proposed adaptive NDN-Edge Computing framework effectively balances latency, reliability, and energy efficiency through a combination of multi-objective optimization, dynamic resource allocation, and robust optimization techniques. These mechanisms allow the system to dynamically adapt to changing network conditions, ensuring optimal performance in IoT environments. The framework formulates latency, reliability, and energy efficiency as a multi-objective optimization problem, ensuring that improving one parameter does not significantly degrade the others. Using the Non-dominated Sorting Genetic Algorithm II (NSGA-II), the system finds a Pareto-optimal balance that maintains low latency, high reliability, and energy efficiency. By leveraging this approach, the framework ensures that real-time applications such as smart surveillance and industrial automation experience minimal delays while maintaining efficient energy usage and robust data transmission. This optimization process is crucial for IoT applications that require a delicate balance between speed, stability, and power consumption. To further enhance this balance, the framework incorporates dynamic resource allocation strategies that respond to real-time network conditions and user demands. One of the key mechanisms is adaptive caching, where frequently requested data is stored at edge nodes to reduce redundant transmissions, thereby minimizing both latency and energy consumption. Additionally, load balancing among edge nodes ensures that computational tasks are evenly distributed, preventing overload on individual nodes and improving overall system reliability. Furthermore, network-aware forwarding strategies are implemented, where data packets are routed based on real-time congestion levels, energy availability, and failure probabilities, ensuring efficient resource utilization. Another critical component of the framework is robust optimization, which allows the system to anticipate and handle uncertainties in IoT environments. Unlike traditional models that assume stable conditions, this framework accounts for fluctuating data demand, dynamic topology changes, and potential node failures. The system dynamically adjusts caching and forwarding decisions to prevent congestion during high-traffic periods. It also reconfigures routing paths in response to edge node failures, ensuring uninterrupted service. Even when up to 30 % of edge nodes fail, the system maintains 95 % reliability, demonstrating its resilience in unpredictable conditions. By integrating stochastic programming, the framework ensures that IoT networks remain stable and efficient despite fluctuating demand and environmental challenges. The effectiveness of the proposed framework is supported by extensive simulation results, which confirm significant improvements over traditional models. The results show a 25 % reduction in latency, which is critical for time-sensitive IoT applications. Additionally, the framework achieves a 15 % reduction in energy consumption by eliminating unnecessary data transmissions and optimizing computational task allocation. Moreover, the cache hit ratio is improved by 20 %, allowing for faster data retrieval and lower network congestion. These improvements collectively demonstrate that the adaptive NDN-Edge Computing framework successfully balances latency, reliability, and energy efficiency, making it a viable solution for large-scale IoT networks.

Altogether, the synthesis of the results discussed in this study proves the efficiency of the offered methodologies in the context of IoT system improvement. The key findings from each methodology are discussed in detail below:

The dynamic resource allocation framework signified a considerable prospect of influencing the manner of conserving the network resources. More particularly it led to minimize the average latency from the case of Peak Traffic at an average of 130 ms to the case in Low Traffic averaging 95ms. The Low Traffic Throughput was determined to be at a maximum of 55 Mbps whilst Energy consumption was reduced to a minimum of 1100 kWh in The Best Optimum. The Cache Hit Ratio was raised to of 70 % under Low Traffic was achieved while the system reliability was consistently above 95 %. It should be noted that the results obtained show that the proposed method allows for providing a balance between performance characteristics and network stability and reliability when the traffic load varies.

The findings of the multi-objective optimization investigation prove that it is possible to obtain a balance of numerous conflicting aims. The employment of different optimization methods like PSO, GA, FO, Hybrid gave variety of solutions. For example, the minimized latency was obtained at 80 ms using FO, and the highest reliability obtained was at 99 % also using FO. As for the power consumption, it remained at its lowest of 950 kWh in the case of the Hybrid solution, and according to the Trade Off Index, FO offered the best trade off with 0. 83. This proves that the type of optimization method chosen can greatly impact on the balance of the performance measures and the integration of the two methodologies present a more balanced optimization solution.

During the uncertain conditions like High Traffic, Node Failure, and Data Surge, the framework of robust optimization did not let the system exceed certain limits. Mean latency was again kept at an acceptable range, reaching its high during Data Surge of 115 ms and the maximum latency remained at 135 ms Through all the conditions, energy efficiency slightly dropped to 75 % while reliability remained at 95 %. Such kind of outcomes confirm the feasibility of the suggested framework to stay on high performance levels even in complicated environment, which demonstrates the applicability of the proposed approach for real world conditions.

Incorporating NDN with the edge computing shown a marked improvement in the efficiency of the overall system. The overall solution established the least of latency at 70 ms, the highest of throughput at 50 Mbps, and the least energy consumption at 900 kWh. Cache hit ratio was at its highest at 75 % which shows that the retrieval of data was effective and the Scalability Index was also the best at 0. 85 proving how the solution is capable of having improved and more demanding networks. Thus, based on these results, it can be concluded that the interaction between NDN and edge computing offers numerous benefits and is a convenient, efficient, and highly reliable system that can be recommended for current IoT applications.

This paper insists on the need to use superior techniques of optimization and integration in the performance of IoT systems. It has been stated that the methodologies provided not only achieve optimisation of the performance indicators but also robustness and expandability required for real-life IoT solutions. Future work will encompass developing better optimization techniques regarding the given system’s performance with consideration being given to more sophisticated solutions such as machine learning. Also, examining how other future technologies, including the 5 G network and blockchain will be incorporated will be important in considering how security, scalability, and efficiency issues will be solved in the next-generation IoT network. It is thus seen that these detailed analyses and key findings manifest the significant weight of the proposed frameworks, which laid requisite groundwork for differential further research and practical applications in the sphere of IoT networking and optimization.

5. Conclusions

This paper gives a clear analysis of the research on the future developments in improving IoT Systems’ performance through power control techniques, multiple objective optimisation, and the combination of NDN and edge computing. These variations show a notable gain over its counterparts in five specific parameters that are latency, throughput, energy consumption, cache hit ratio and scalability. The dynamic resource allocation framework effectively managed the allocated network resources, therefore provided high system reliability and excellent traffic management of the acquired data. The systematic Multi-Objective optimization approach which included various methodologies like Particle Swarm Optimization, Genetic Algorithm, Fuzzy Optimization, and Hybrid strategies helped to get an optimal answer of latency, reliability, and energy consumption with less trade-off of each other. Thus, load balancing and robust optimization provided good results in controlling the system performance in conditions of high traffic, node failures, and data bursts, which confirmed the reliability of the presented frameworks. The integration considerations of NDN with the edge computing highlighted that, the overall performance of the network in terms of latency, energy consumption was improved and the random throughput and cache hit ratios were boosted. This integration is supposed to provide a versatile solution that would be able to cover the expanding requirements of contemporary IoT deployments. Overall, the observed results prove the ability of these methodologies to transform the IoT systems in terms of efficiency, robustness, and scalability. Future work will be directed at developing more detailed and advanced optimization procedures with reference to the application of such approaches as machine learning and artificial neural networks, in order to continue improving the effectiveness of the systems in question. Furthermore, investigating the possibilities of integrating such a innovative technologies as 5 G and blockchain in the Internet of Things networking will play a significant role in overcoming future challenges, especially aspects in security and extendibility. The current study presents a good ground for aspiring subsequent research and viable application of the IoT technologies in the future as it encourages the improvement of IoT technologies.

CRediT authorship contribution statement

Ahmed M. Alwakeel: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Ahmed .M Alwakeel reports a relationship with University of Tabuk that includes: employment and funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work is supported by a research grant from the Research, Development, and Innovation Authority (RDIA), Saudi Arabia, grant no. 13010-Tabuk-2023-UT-R-3–1-SE.

Acknowledgments

This Study could not be started, nor achieved without University of Tabuk encouragement, and its continued support.

Data availability

The author used data to support the findings of this study that is included within this article.

References

[1] Wireless communications and mobile computing - 2022 - karim - retracted a survey on information-centric networking w, 2022.
[2] S. Bouzefrane, S. Banerjee, A. Abane, M. Daoui, P. Muhlethaler, A realistic deployment of named data networking in the internet of things, J. Cyber Secur. Mobil. 9 (1) (2020).
[3] N.A. Askar, et al., Forwarding strategies for named data networking based iot: requirements, taxonomy, and open research challenges, IEEE Access 11 (August 2023) 78363–78383.
[4] Y. Guo, M.K. Afzal, B.S. Kim, C. Zhang, B. Sonkoly, Ieee access special section editorial: information centric wireless networking with edge computing for 5 *g* and IoT, IEEE Access 8 (2020) 139737–139740.
[5] T.C. Schmidt, C. Gundogan, P. Kietzmann, M. Wahlisch, Icnlowpan-named-data networking for low power IoT networks, in: 2019 IFIP Networking Conference, 2019.
[6] R. Hassan, K. Nisar, F.A. Karim, A.H.M. Aman, M. Uddin, Named data networking: a survey on routing strategies, IEEE Access 10 (July 2022) 90254–90270.
[7] F.A. Feltus, S. Ficklin. Session 1: ndn trial deployments publishing genomics datasets into ndn testbed and integrating with data-centric ecosystems for large-scale data-intensive science, 2019, pp. 1–16.
[8] M.A. Hail, Iot-ndn: an IoT architecture via named data networking (ndn), in: 2019 IEEE International Conference on Industrial 4.0, Artificial Intelligence, and Communications Technology, 2019, pp. 74–80, pages.
[9] P. Hao, X. Wang, Integrating phy security into ndn-iot networks by exploiting mec: authentication efficiency, robustness, and accuracy enhancement, IEEE Transac. Sign. Inform. Proc. Netw. 5 (4) (2019) 792–806.
[10] A. Gazdar, L. Alkwai, A. Belghith, S. AlAhmadi, Awareness of user mobility in named data networking for iot traffic under the push communication mode, J. Netw. Comp. Applic. 213 (2023).
[11] B. Wissingh, N. Suri, M. Hauge, L. Campioni, M. Tortonesi, L. Landmark, Experimental evaluation of named data networking (ndn) in tactical environments, in: Proceedings - IEEE Military Communications Conference, volume 2019-November, 2019.
[12] C. Senna, L. Gameiro, M. Lú ıs, Ndniot-fc: IoT devices as first-class traffic in name data networks, Fut. Int. 12 (11) (2020) 1–21.
[13] A. Iera, M. Amadeo, C. Campolo, A. Molinaro, Named data networking for iot: an architectural perspective, in: EuCNC 2014 - European Conference on Networks and Communications, 2014 6882665 page.
[14] A. Molinaro, M. Amadeo, C. Campolo, G. Ruggeri, Iot data processing at the edge with named data networking, in: 24th European Wireless 2018, 2018, pp. 38–43, pages.
[15] G. Ruggeri, M. Amadeo, C. Campolo, A. Molinaro, Beyond edge caching: freshness and popularity aware iot data caching via ndn at internet-scale, IEEE Transac. Green Commun. Network. 6 (1) (2022) 352–364.
[16] A. Belghith, T. Monteil, K. Drira, M. Meddeb, A. Dhraief, S. Al-Ahmadi, Named data networking: a promising architecture for the internet of things (IoT), Int. J. Semantic Web Infor. Syst. 14 (2) (2018) 86–112.
[17] P. Mekbungwan, PhD thesis, Sorbonne Universit ́es, 2022.
[18] Y. Meng, A. Bin Ahmad, Performance measurement through caching in named data networking based internet of things, IEEE Access 11 (June 2023) 120569–120584.
[19] W. Muhammad, et al., The emerging of named data networking: architecture, application and technology, IEEE Access (2023) 1. PP.
[20] M.A. Naeem, R. Ullah, S. Chudhary, Y. Meng, A critical analysis of cooperative caching in ad hoc wireless communication technologies: current challenges and future directions, Sensors 25 (2025). MDPI[Online]. Available, https://www.mdpi.com.
[21] P. Pathak and S. Weber. Practical implementation of named data networking for internet of things, August 2022.
[22] D. Florou, V. Demiroglou, M. Martinis, V. Tsaoussidis, Iot data collection over dynamic networks: a performance comparison of ndn, dtn and nod, in: 2023 IEEE World Forum on Internet of Things, 2023.
[23] R. Hassan, W.M.H. Azamuddin, A.H.M. Aman, N. Mansor, Comparison of named data networking mobility methodology in a merged cloud internet of things and artificial intelligence environment, Sensors 22 (17) (2022).
[24] E. Olsen. A review of research works in named data networking and information centric networking, 2020.
[25] W. Shang, et al., Named data networking of things, in: 2016 IEEE First International Conference on Internet-of-Things Design and Implementation, 2016, pp. 117–128, pages.
[26] A. Almogren, I. Ud Din, C. Maple, Z. Ali, M.A. Shah, H.A. Khattak, Named data networking for efficient iot-based disaster management in a smart campus, Sustainability 12 (8) (2020) 1–21.
[27] X. Wang, X. Wang, Y. Li, Ndn-based iot with edge computing, Fut. Gener. Comp. Syst. 115 (2021) 397–405.