TÜRKİYE CUMHURİYETİ
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL
ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT

A MICRO NAMED DATA NETWORKING FRAMEWORK FOR INTERNET OF THINGS OVER IEEE 802.15.4

SEDAT BİLGİLİ

Ph.D.

ADANA 2024

TÜRKİYE CUMHURİYETİ
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL
ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT

A MICRO NAMED DATA NETWORKING FRAMEWORK FOR INTERNET OF THINGS OVER IEEE 802.15.4

SEDAT BİLGİLİ

Ph.D.

THESIS ADVISOR
ASSOC. PROF. DR. ALPER KAMİL DEMİR

ADANA, 2024

ÖZET

IEEE 802.15.4 ÜZERİNDE NESNELERİN İNTERNETİ İÇİN MİKRO ADLANDIRILMIŞ VERİ AĞI YAPISI

Sedat BİLGİLİ
Doktora, Elektrik-Elektronik Anabilim Dalı
Danışman: Doç. Dr. Alper Kamil DEMİR

Ocak 2024, 122 sayfa

Nesnelerin İnterneti (IoT), popülerliğini gün geçtikçe artıran bir konsepttir. Günlük hayatta artık sıklıkla karşılaşmaya başladığımız bu IoT cihazları, sağlık, akıllı ev/ofis, tarım, otomasyon gibi pek çok alanda kullanılmaktadır. IoT ağları çoğunlukla bu IoT cihazlarından veri almaya dayalıdır. Bu veri-merkezcil yaklaşım, günümüzde popülaritesi oldukça artan Veri Merkezli Ağ (ICN) mimarilerinden birisi olan Adlandırılmış Veri Ağları'nın (NDN) veri-merkezciliği ile örtüşmektedir. Bunun sonucunda, IoT konseptinin ve NDN mimarisinin birleşimi, NDNoT konseptini ortaya çıkartmıştır.

Bu çalışma, oldukça yeni olan NDNoT konsepti üzerindeki gelişmelere odaklanırken, bu konsept ile uyumlu olan µNDN protokol yığınını ve yenilikçi IfNoT mekanizmasını önermektedir. Çalışma, bunların yanı sıra, TM-RONR iletim mekanizmasını da literatüre sunmaktadır. Gerçekleştirilen bu çalışmaların yanı sıra, µNDN yığını üzerinde etkisi olabilecek önbellekleme, FIB boyutu gibi kavramlar da incelenmiş ve performans analizlerine tabi tutulmuştur.

Anahtar Kelimeler: nesnelerin interneti, adlandırılmış veri ağları, gömülü sistemler, ieee 802.15.4, kablosuz algılayıcı ağları, saldırı engelleme.

ABSTRACT

A MICRO NAMED DATA NETWORKING FRAMEWORK FOR INTERNET OF THINGS OVER IEEE 802.15.4

Sedat BİLGİLİ
Ph.D., Department of Electrical and Electronic Engineering
Supervisor: Assoc. Prof. Dr. Alper Kamil DEMİR

January 2024, 122 pages

The Internet of Things (IoT) is a concept that is increasing in popularity day by day. These IoT devices, which we increasingly encounter daily, are used in many areas, such as health, smart home/office, agriculture, and automation. IoT networks are primarily based on receiving data from these IoT devices. This data-centric approach aligns with the data-centricity of Named Data Networks (NDN), one of the Information Centric Network (ICN) architectures increasing in popularity today. As a result, the combination of the IoT concept and NDN architecture gave rise to the NDNoT concept.

While this study focuses on improvements in the relatively new NDNoT concept, it proposes the μNDN protocol stack and the innovative IfNoT mechanism on top of μNDN. In addition, the study also presents the TM-RONR forwarding mechanism to the literature. Along with these studies, concepts such as caching and FIB size, which may impact the μNDN stack, were also examined comprehensively and subjected to performance analysis.

Keywords: internet of things, named data networking, embedded systems, ieee 802.15.4, wireless sensor networks, attack mitigation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

6LoWPAN	: IPv6 over Low-Power Wireless Personal Area Network(s)
CCN	: Content-Centric Networking
CDN	: Content Delivery Network
CoAP	: Constrained Application Protocol
CS	: Content Store
CSMA/CA	: Carrier Sense Multiple Access / Collision Avoidance
DDoS	: Distributed Denial of Service
FIA	: Future Internet Architecture
FIB	: Forwarding Information Base
HTTP	: Hyper Text Transfer Protocol
ICN	: Information Centric Networking
IEEE	: Institute of Electrical and Electronics Engineers
IETF	: Internet Engineering Task Force
IFA	: Interest Flooding Attack
IP	: Internet Protocol
IoT	: Internet of Things
KB	: Kilo Bytes
MAC	: Media Access Control
MANET	: Mobile Adhoc Network
MB	: Mega Bytes
MBF	: Must be Fresh
MQTT	: Message Queuing Telemetry Transport
NDN	: Named Data Networking
NDNoT	: Named Data Networking of Things
PIT	: Pending Interest Table
QoS	: Quality of Service
RONR	: Reactive Optimistic Name-based Routing
SDN	: Software Defined Networking
TCP	: Transmission Control Protocol
TTL	: Time to Live

UDP	: User Datagram Protocol
VIF	: Vanilla Interest Flooding
WANET	: Wireless Adhoc Network
WSN	: Wireless Sensor Network
μNDN	: Micro Named Data Networking

LIST OF SYMBOLS

α : IfNoT alpha

min_α : IfNoT Minimum alpha

ψ : IfNoT Increase Factor

λ : IfNoT Decrease Factor

1. INTRODUCTION

1.1. Internet of Things

The Internet, a sophisticated global network that revolutionized communication and connectivity, has continually evolved since its inception. Initially conceived as a means of interconnecting computers for data exchange, the Internet has transcended its original purpose, becoming an indispensable part of modern life. The emergence of the Internet dates back to the late 20th century, when pioneering technological advancements enabled the interconnection of computers across vast distances. This interconnectivity transformed communication, commerce, and information dissemination, laying the groundwork for a globally interconnected world. The evolution of the Internet has led to a groundbreaking concept known as the Internet of Things (IoT) (Li, Xu, and Zhao 2015; Rose, Eldridge, and Chapin n.d.; Tan and Wang 2010). This idea goes beyond linking regular computers and connects many smart devices, as shown in Figure 1.1.1. These devices, equipped with sensors and connectivity, form a network where everyday objects interact, share information, and perform tasks, revolutionizing how we live and work. The IoT paradigm extends the Internet's reach to encompass everyday objects, equipping them with sensing, computing, and communication capabilities. This interconnected network of 'smart' devices facilitates seamless data exchange and enables a spectrum of applications across diverse domains, including healthcare (Kulkarni and Sathe n.d.; YIN et al. 2016), agriculture (Tzounis et al. 2017; Verdouw, Wolfert, and Tekinerdogan 2016), transportation (Liu 2012; Patel, Narmawala, and Thakkar 2019), and industrial automation (Breivold and Sandström 2015; Deshpande et al. 2016). Within the IoT ecosystem, devices vary widely, ranging from sensors and actuators to wearable gadgets and industrial machinery. These devices generate massive amounts of data, leading to new challenges and opportunities in managing, processing, and utilizing this information effectively.

Figure 1.1.1. An Illustration of Interconnected Devices within the Concept of IoT (Internet of Things)

While the Internet of Things (IoT) revolutionizes connectivity among various smart devices, the backbone of this interconnected landscape relies significantly on specialized communication protocols. These protocols act as the communication highways that enable seamless interaction and data exchange between IoT devices across diverse environments. One such essential protocol, the IEEE 802.15.4 (Baronti et al. 2007; Gutierrez et al. 2001) standard, plays a foundational role in facilitating communication within low-power and low-data-rate wireless personal area networks. This protocol serves as the base for numerous IoT applications, providing the framework for connectivity, data transfer, and network formation among IoT devices. In addition to IEEE 802.15.4, various other protocols, including Bluetooth Low Energy (BLE) (Gomez, Oller, and Paradells 2012), Zigbee (Ramya, Shanmugaraj, and Prabakaran 2011), and LoRaWAN (Haxhibeqiri et al. 2018), cater to different IoT deployment scenarios, each offering distinct advantages in terms of range, power consumption, data rate, and network scalability. These specialized protocols serve as the arteries of the IoT ecosystem, enabling devices to communicate, collaborate, and function harmoniously within interconnected networks. Understanding and optimizing these protocols are crucial for enhancing the efficiency, reliability, and security of IoT communication in diverse environments.

At the heart of the IoT's communication infrastructure, the IEEE 802.15.4 standard stands as a foundational protocol facilitating connectivity in low-power wireless personal area networks. In accordance with the established protocol, the 6LoWPAN (Mulligan 2007) adaptation layer assumes a critical function in the effective transmission of IPv6 (Deering and Hinden 1995) packets across networks operating under the IEEE 802.15.4 standard. By optimizing packet sizes and compressing headers, 6LoWPAN addresses the constraints of IoT devices, ensuring efficient communication within resource-constrained environments. At the transport layer, while TCP (Transmission Control Protocol) (Postel 1981) offers robustness and reliability in communication, its heavyweight nature can be impractical for resource-constrained IoT devices. The extensive overhead and reliability mechanisms of TCP may strain the limited computational resources and energy capacities of these devices. As an alternative, UDP (User Datagram Protocol) (Postel 1980), known for its lightweight and connectionless nature, becomes a favorable choice for constrained IoT devices. Unlike TCP, UDP's reduced overhead and simplicity in handling data make it suitable for scenarios prioritizing low latency and quick data transmission over absolute reliability. Many IoT applications opt for UDP due to its ability to efficiently manage data with minimal processing requirements, making it a preferred protocol in resource-constrained environments.

Beyond the foundational protocols operating at lower layers, the application layer plays a pivotal role in defining how IoT devices interact and exchange information. Within the IoT landscape, various application layer protocols and frameworks, such as MQTT (Message Queuing Telemetry Transport) (Soni and Makwana 2017) and CoAP (Constrained Application Protocol) (Shelby, Hartke, and Bormann 2014) coordinate communication and data exchange among devices.

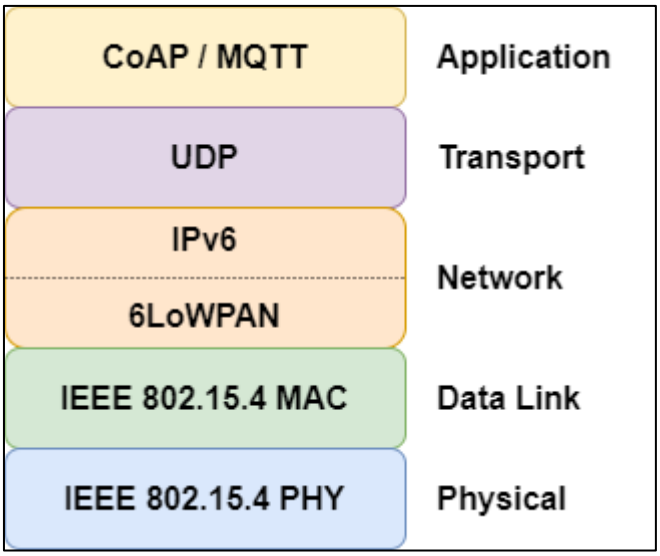


Figure 1.1.2. Generic 6LoWPAN Stack for Constrained Internet of Things Devices

This convergence of protocols forms a layered stack designed for constrained environments, as shown in Figure 1.1.2. At the physical layer, the IEEE 802.15.4 PHY provides the underlying radio communication, while the IEEE 802.15.4 MAC operates in the data link layer. In the network layer, IPv6 and 6LoWPAN work in cooperation, optimizing packet delivery for constrained devices. Moving up the stack, UDP operates at the transport layer, ensuring lightweight communication. Finally, at the application layer, CoAP or MQTT orchestrate messaging suited for constrained IoT scenarios. Even optimized protocols like IPv6, 6LoWPAN, UDP, CoAP, MQTT, and others, tailored for efficiency in constrained IoT environments, can still introduce complexities and overhead.

1.2. Information Centric Networking

Information Centric Networking (ICN) (Ahlgren et al. 2012a; Xylomenos et al. 2014a) stands as an evolution in network architecture, representing a difference from traditional host-centric models towards a focus on content dissemination and retrieval. Its origins trace back to the changing landscape of network demands, where the demand moved from connectivity to content delivery. This transition paved the way for the conceptualization of ICN's architecture, which diverges from IP address-based communication to a content-centric framework. Here,

content is identified through unique hierarchical names, enabling seamless retrieval and reducing dependence on specific endpoints.

ICN's fundamental design encompasses decentralization, enhancing network robustness and resilience. Leveraging strategically deployed caching mechanisms, ICN optimizes content distribution, mitigating issues like network congestion and latency. This decentralized approach not only enhances performance but also minimizes redundant data transmission, optimizing resource utilization and fostering greater network efficiency.

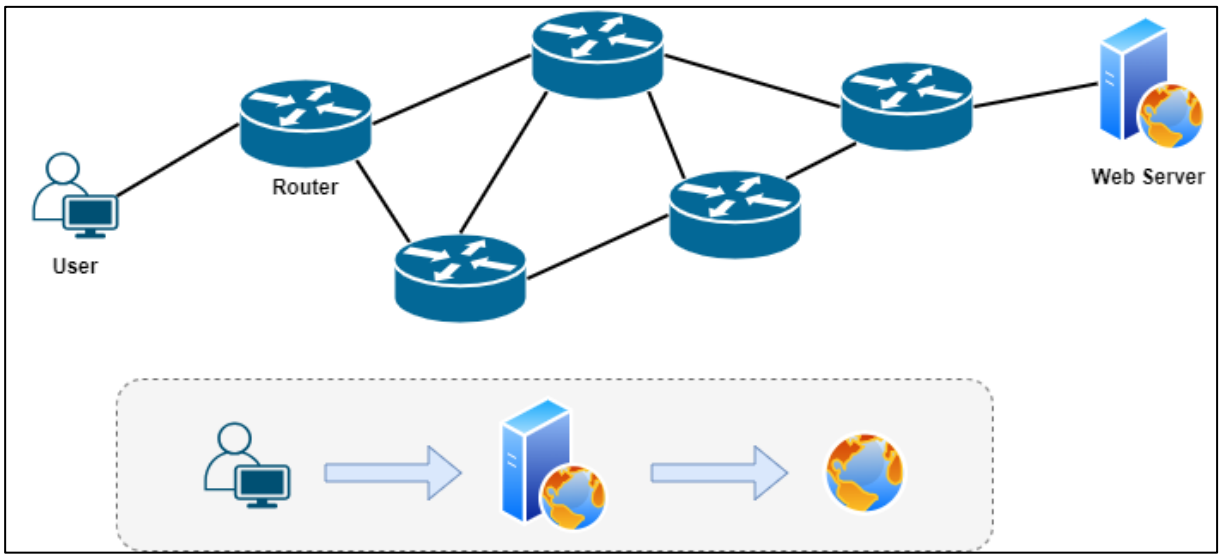


Figure 1.2.1. Information (Content) Retrieval on a Traditional Networks with Host Addressing

As shown in Figure 1.2.1 and Figure 1.2.2, ICN changes the network's abstraction from "named hosts" to "named content". ICN utilizes the memory of routers for content caching. ICN focuses on securing the content, not the host itself.

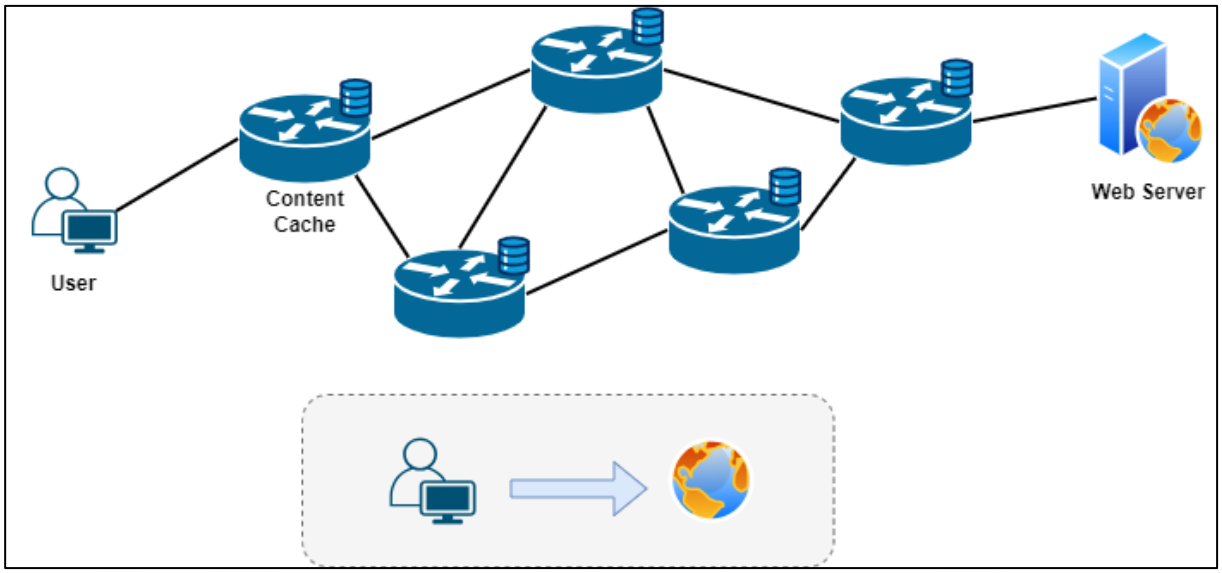


Figure 1.2.2. Information (Content) Retrieval on an Information Centric Networks with Data Addressing

Moreover, ICN features an inherent security model integrated into its architecture. Employing cryptographic protocols and integrity verification mechanisms, ICN ensures data authenticity and confidentiality during content transmission. This robust security framework provides trust and reliability, which are crucial in diverse network environments and for safeguarding information integrity.

In the broader context of Future Internet Architectures (FIAs) (Anon n.d.), ICN emerges as a milestone in reshaping data communication landscapes. Its alignment with emerging technologies, such as edge computing and distributed systems, positions ICN at the forefront of network innovation. Integration of ICN principles into FIAs promises avenues for advancing distributed computing, real-time data processing, and content delivery networks.

As research and collaborations progress, ICN undergoes continual refinement of protocols and implementations. This evolution underscores its significance in shaping the future of network infrastructures. ICN brings together efficiency, security, and adaptability, setting the stage for an era where the focus on content drives the evolution of networks, introducing a new standard for connectivity and data dissemination.

Furthermore, ICN's growth sparks interest across academia and industry, fostering interdisciplinary exploration. Its potential applications span various domains, including IoT, multimedia streaming, edge computing, and content delivery networks. This expansive reach amplifies ICN's promise in revolutionizing network infrastructures and establishing a more efficient, secure, and adaptable approach to data communication.

ICN's evolution drives innovation, tending to collaborate to refine its architecture and expand applications. ICN's integration transforms connectivity and information sharing in a content-centric digital ecosystem.

1.3. Named Data Networking

Named Data Networking (NDN) (Saxena et al. 2016; L. Zhang et al. 2014) emerges as a transformative networking paradigm derived from the foundational concepts of Information-Centric Networking (ICN) (Ahlgren et al. 2012a; Xylomenos et al. 2014a), notably Content-Centric Networking (CCN) (Amadeo, Campolo, et al. 2014; Shinde and Chaware 2018). This evolution marks a departure from traditional communication models, redefining the way information is accessed and disseminated within networks. Derived from the principles of ICN, NDN expresses a shift from the conventional host-centric communication model to a content-centric approach. This fundamental reimagining of data retrieval places emphasis not on where data is stored but rather on uniquely naming the data itself. Within NDN, data is addressed using human-readable names, facilitating a network where content retrieval is based on the inherent identity of information rather than its location. As shown in the hourglass model in Figure 1.3.1, the thin waist for the traditional approach is IP packets, while the narrow waist for the NDN approach is content chunks.

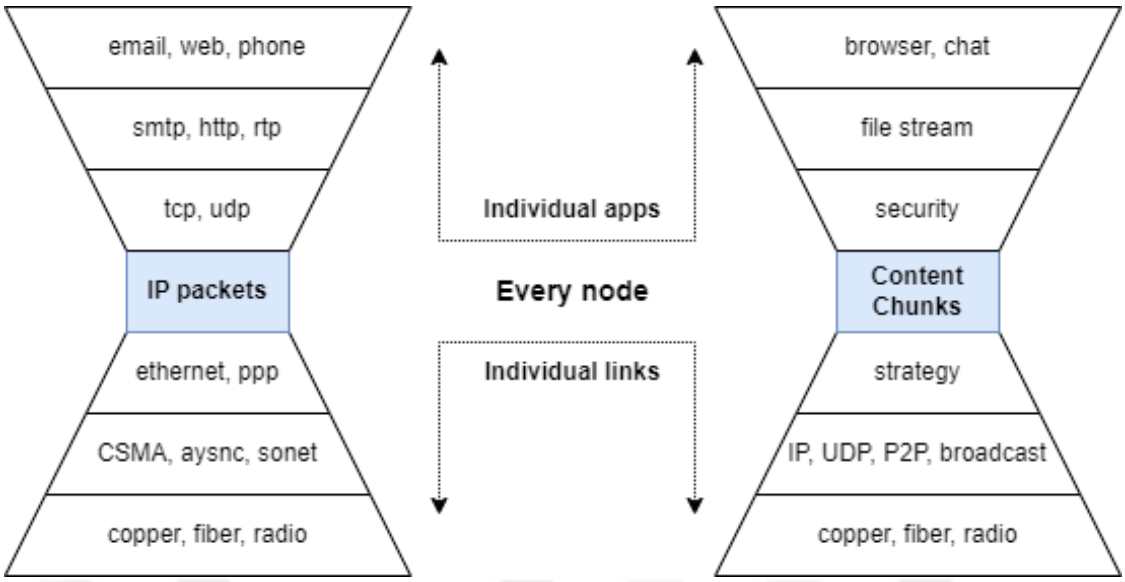


Figure 1.3.1. IP Approach and ICN/NDN Approach Comparison with an Hourglass Model with their Narrow Waists

At its core, NDN operates on a simple yet innovative principle: expressing interest in specific data. Devices within an NDN-enabled network express interest in desired content, prompting the network to deliver the requested data. This Interest-Data exchange mechanism serves as the backbone of NDN, enabling intuitive and efficient content retrieval across the network.

The architectural design of NDN aligns with this content-centric model. NDN forwarders and specialized network nodes intelligently route Interest packets and cache Data packets, optimizing the dissemination and retrieval of content. Moreover, NDN incorporates robust security measures, ensuring the authenticity and integrity of data through mechanisms such as data signing and trust models.

NDN's content-centric approach transforms data access, enabling efficient information retrieval across diverse applications and environments. Whether retrieving sensor data from various sources in IoT ecosystems or accessing multimedia content in distributed networks, NDN's content-centricity fosters an intuitive and adaptable network environment. An example use case arises in healthcare settings where patient data retrieval is critical. With NDN, medical practitioners express interest in 'patient records' without needing to pinpoint the exact storage

locations. This approach ensures quick access to the most relevant and up-to-date patient information, enhancing the efficiency of healthcare delivery without concerns about data location specifics.

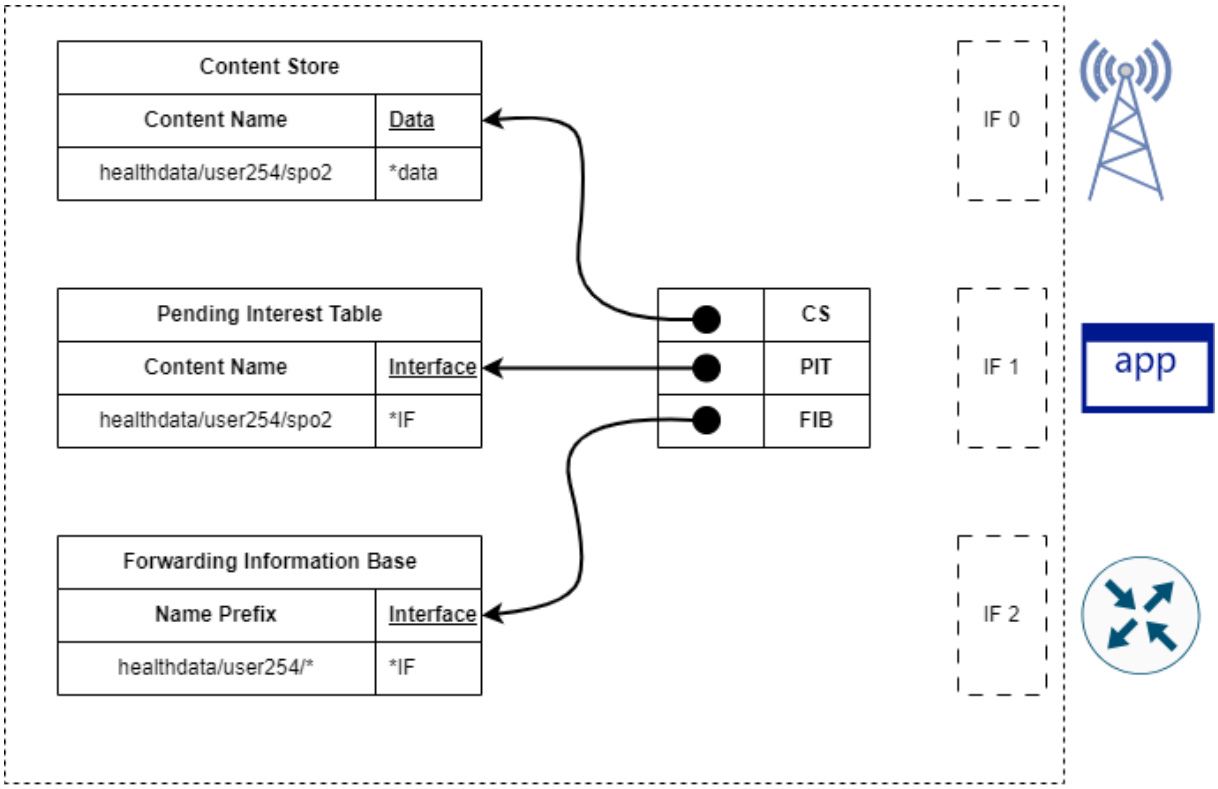


Figure 1.3.2. An NDN Router with its Components: Content Store, Pending Interest Table, Forwarding Information Base, and Interfaces

Devices within an NDN-enabled network articulate their interest in desired content by sending Interest packets containing the name of the data they seek. These Interest packets traverse the network, searching for the requested content based solely on its unique identifier. For instance, a device seeking ‘weather updates’ generates an Interest packet with the corresponding data name. The network’s forwarding infrastructure, equipped with NDN forwarders, intelligently routes these Interest packets toward the data sources or repositories that might contain the requested content. As these packets progress through the network, NDN forwarders examine the packet names and attempt to satisfy the Interest by locating the requested data. Upon encountering a match between an Interest and available data, the network delivers the corresponding Data packet back to the requesting device. This Data packet contains the

requested content, fulfilling the initial expression of interest. One of the striking attributes of the Interest-Data exchange mechanism is its efficiency in content retrieval. Rather than necessitating specific knowledge of data locations, devices express interest based on content names, allowing for decentralized and dynamic retrieval. This approach facilitates flexible adaptation to changing data sources or distribution methods without relying on fixed addresses or locations.

A generic NDN router (node), as shown in Figure 1.3.2, has three database types and communication interfaces. PIT (Pending Interest Table) is a crucial component of the forwarding plane. It's a temporary table used by routers to store incoming interests for content that needs to be fetched. When a router receives an interest packet for a specific content, it checks its PIT to determine if it has already forwarded an interest for that content. If not, it adds an entry to the PIT and forwards the interest toward the content source. The FIB (Forwarding Information Base) is another key element in NDN's forwarding process. It functions as a routing table that contains information about which interface to use for forwarding packets. Unlike traditional IP-based routing tables, the FIB in NDN is typically based on content names or prefixes, mapping them to outgoing interfaces or next hops. When an interest arrives at a router, the FIB helps determine the next hop toward the content's location based on the content's name or prefix in the interest packet. CS (Content Store), often referred to as a cache, is a storage component within NDN routers that temporarily holds recently requested content. It allows routers to fulfill content requests locally without needing to fetch the content from the original source. When a router receives content, it may store it in the CS based on caching policies. Subsequent requests for the same content can then be satisfied directly from the local cache, reducing latency and conserving network bandwidth. An NDN router can have multiple Interfaces. These interfaces may be communication connections such as radio or ethernet, or they may be upper-layer applications.

NDN architecture can operate on top of physical and data link layers with its own protocol stack. In addition, the NDN architecture can also run on a generic TCP/IP or generic 6LoWPAN protocol stack. In other words, the NDN architecture doesn't require a specific protocol stack like TCP/IP or 6LoWPAN to function, but it is capable of operating on top of these protocol

stacks. This ensures that the NDN architecture can be easily used and migrated over existing networks.

1.4. Named Data Networking of Things

In the realm of IoT, where resource-constrained devices play a pivotal role, the concept of Named Data Networking (NDN) introduces a paradigm shift in data handling and communication. Constrained IoT devices, often limited in computational power, memory, and energy resources, pose significant challenges for traditional communication protocol stacks like TCP/IP or 6LoWPAN stacks. As mentioned in the previous section, the NDN protocol stack does not depend on the presence of a TCP/IP or 6LoWPAN protocol stack. Thus, NDN architecture consumes less system resources on constrained IoT devices. Embracing NDN within the IoT ecosystem introduces the concept of NDNoT (Aboodi, Wan, and Sodhy 2019a; Hail 2019a; Z. Zhang et al. 2018), showcasing the powerful integration of Named Data Networking within the Internet of Things, revolutionizing how data is accessed and exchanged among constrained IoT devices.

NDN's architecture, rooted in a more data-centric approach, proves to be inherently suitable for these constrained IoT environments. Unlike conventional communication models that revolve around addressing and transporting data based on location, NDN's emphasis on content retrieval through naming data itself serves as a revolutionary solution. One of the primary advantages of NDN for constrained IoT devices is its independence from reliance on specific protocol stacks like TCP/IP or 6LoWPAN. While these traditional stacks operate with their limitations, NDN, with its content-centric nature, streamlines data retrieval and communication, bypassing the complexities and overhead often associated with these stacks.

The inherent efficiency of NDN's content-centric model aligns seamlessly with the core requirements of IoT devices that prioritize data exchange, quick retrieval, and optimized resource usage. By expressing interest in specific content rather than pointing storage locations, NDN allows for decentralized, dynamic retrieval without the need for fixed addresses or locations, a crucial aspect of the constantly evolving nature of IoT networks. The data-centricity of NDN not only streamlines data retrieval but also fosters adaptability to changing data sources

and distribution methods, a vital characteristic in the rapidly evolving landscape of IoT applications.

In the domain of IoT and Named Data Networking (NDN), a notable gap exists in the availability of an open-sourced NDN protocol stack specifically tailored for IoT devices. Addressing this gap, this study introduces µNDN, an NDN protocol stack specially implemented for IoT/NDNoT devices operating on the Contiki NG operating system (Oikonomou et al. 2022). The introduction of µNDN paves the way for addressing this gap, aiming to fill this field by providing a specialized, open-source NDN protocol stack optimized for resource-constrained IoT environments.

1.5. Proposed NDNoT Protocol Stack: µNDN

Within the domain of constrained IEEE 802.15.4 IoT devices, the absence of an open-source NDN stack tailored to these specific constraints is notable. As far as current knowledge extends, the emergence of the proposed µNDN Protocol Stack (Bilgili and Demir 2022; Demir and Bilgili 2022) fills this critical void. Developed to cater specifically to the limitations inherent in IEEE 802.15.4 IoT environments, µNDN stands as a pioneering open-source solution for these constrained networks.

Operating within the Contiki NG operating system, µNDN is uniquely designed to address the resource limitations prevalent in IoT devices, conforming to the strict requirements of IEEE 802.15.4 standards. Its adaptability and modularity stand as pillars supporting future improvements and further developmental studies within this domain. The µNDN Protocol Stack's modular architecture fosters an environment conducive to ongoing enhancements, enabling seamless support for advancements and tailored developments in this specialized realm of IoT networking.

Developed as an implementation of the NDN architecture, µNDN focuses on enabling efficient communication within constrained IoT environments. Embracing the foundational principles

of NDN while accommodating the inherent limitations of devices operating under IEEE 802.15.4 standards, this protocol stack emphasizes optimized data transmission, forwarding strategies, and resource utilization. Key features of the µNDN Protocol Stack encompass its header structure, naming conventions, and packet compositions for both Interest and Data packets. The architecture optimizes these elements to operate within the confined data transmission parameters of 802.15.4 networks, effectively maximizing the available payload while complying with stringent packet size restrictions.

Additionally, the µNDN Protocol Stack integrates forwarding mechanisms such as VIF (Vanilla Interest Flooding) and RONR (Reactive Optimistic Name-based Routing) to efficiently manage data transmission within the network. These strategies have been ported and adapted to suit the limitations and operational constraints of IoT devices under the 802.15.4 standard.

The evaluation and validation of the µNDN Protocol Stack have been conducted both in simulation environments and on real IEEE 802.15.4 hardware. The thorough analyses conducted across various network topologies and metrics provide comprehensive insights into the functionality, efficiency, and comparative performance of forwarding mechanisms, paving the way for its potential implementation in constrained IoT networks. The proposed NDNoT Protocol Stack, referred to as µNDN, has been detailed in the preceding sub-section, "µNDN Protocol Stack", under the "Materials and Methods" section.

1.6. Background of µNDN Framework Studies

In the context of µNDN Framework Studies, particularly within the evolving field of NDNoT, a range of critical mechanisms and strategies have been developed and analyzed. These studies are pivotal in addressing the unique challenges presented by NDNoT environments, such as constrained resources, high-density networks, and the need for efficient data dissemination. The following subsections delve into the core components of the µNDN framework, beginning with the forwarding mechanisms. These components are essential in ensuring that NDNoT not only

meets the demands of modern network traffic but also leverages its distinctive content-centric networking paradigm to optimize performance and security in IoT ecosystems.

1.6.1. Forwarding Mechanisms

Forwarding mechanisms in NDNoT involve routing data requests based on content names rather than source or destination addresses. In this system, when a device requests data (via an Interest packet), this request is forwarded through the network based on the name of the data until it reaches a node that has the data. The data is then sent back along the reverse path to the requester. Nodes in the network can store data they've seen to quickly respond to future requests for the same data, enhancing efficiency and reducing latency. The entire process should be optimized for the low-power, high-density nature of NDNoT environments.

1.6.2. Interest Flooding Attack Mitigation

In NDNoT, Interest Flooding Attack mitigation involves several strategies to prevent the overload of network resources due to maliciously excessive requests for data. These strategies typically include monitoring and controlling the rate of Interest packet forwarding, dynamically adjusting forwarding strategies based on network traffic patterns, and utilizing mechanisms to detect and block suspiciously high volumes of Interest requests from specific nodes. Additionally, employing intelligent caching and collaborative security approaches where nodes work together to identify and mitigate attack patterns also plays a crucial role. These measures are essential in preserving the efficiency and reliability of NDNoT, especially given the resource-constrained environments in which it often operates.

1.6.3. Caching

Caching in Named Data Networking of Things (NDNoT) is a critical feature that enhances the network's efficiency and responsiveness. In NDNoT, nodes store frequently requested data in

their local caches. When a data request (Interest packet) is received, a node first checks its cache; if the requested data is available, it is immediately sent back, significantly reducing data retrieval time and network traffic. This local caching is particularly beneficial in IoT environments, where devices often request the same data repeatedly (e.g., sensor readings). It also aids in network scalability by distributing the load across multiple nodes and reduces dependency on the original data source, leading to better bandwidth utilization and lower latency. Caching in NDNoT is dynamic, with data being stored based on popularity and access patterns, and it's optimized to suit the limited storage capacities typical of IoT devices.

1.6.4. FIB/PIT Sizing

In Named Data Networking of Things (NDNoT), managing the size of the Forwarding Information Base (FIB) and the Pending Interest Table (PIT) is a balancing act. These structures need to be large enough to efficiently handle network demands but small enough to fit within the limited memory resources typical of IoT devices. The FIB, which directs where to forward data requests, and the PIT, which keeps track of outstanding requests, must be optimized to ensure quick and effective data routing while avoiding memory overload.

2. LITERATURE REVIEW

Within the evolving landscape of the IoT, the integration of NDN stands as a crucial advancement, signifying a pivotal move towards communication paradigms that prioritize data. This section is dedicated to navigating the intersection between the IoT and NDN within the framework of the IEEE 802.15.4 standard. The seminal works, foundational theories, and contemporary research in this domain are summarized in this section, offering a comprehensive exploration of the groundwork that underlays the prospective μNDN framework. Through an in-depth analysis of existing knowledge, this section seeks to clarify the potential, challenges, and future directions and pathways within the dynamic realm of IoT networking transformation.

2.1. Information Centric Networking

Information-Centric Networking (ICN) (Ahlgren et al. 2012b; Xylomenos et al. 2014b) represents a transformative shift in networking paradigms, emphasizing content accessibility over data location. Information distribution plays a significant role in today's Internet landscape. Alongside web-based content distribution, the emergence of alternative technologies like P2P (Peer-to-Peer) and CDN (Content Delivery Networks) has facilitated a communication model centered on accessing data by name, irrespective of the location of the original server. A study (Kurose 2014) discusses the evolution of communication networks from circuit-switched networks to packet-switched networks and ultimately to today's ICNs. It notes that while ICNs share features with their predecessors, they also have many unique characteristics of their own, marking a significant evolution in the field of networking.

The inception of ICN was primarily driven by the limitations of the host-centric internet to handle scenarios posed by the current internet, including web applications, multimedia streaming, IoT, and more. ICN, focusing on named data rather than data locations, has shown promise in more efficient data access through in-network caching, simplified content request messages, and content-specific security functions. Various organizations and research groups are advancing this novel architecture (Firdhous 2017; Yu et al. 2019).

One survey study (Xylomenos et al. 2014c) provides a comprehensive survey identifying the core functionalities of ICN architectures, their key proposals, and the main unresolved research challenges in this area, highlighting the shift in internet usage from host-centric to content-centric models. Similarly, another study (Carofiglio et al. 2015b) discusses the potential of ICN in scalable mobile backhauling, identifying significant opportunities for latency reduction and network cost savings. Another survey (Ngaffo, El Ayeb, and Choukair 2020) presents the ICN concept, its key features, different architectures, and the challenges and opportunities in service discovery, emphasizing the shift towards content-centric approaches.

The integration of ICN with other emerging networking paradigms, such as Software Defined Networking (SDN), has also been a subject of interest. The paper (Q. Y. Zhang et al. 2018) provides a survey on integrating SDN and ICN, discussing the strengths, opportunities, and

potential security benefits of this integration. Also, it emphasizes the practical implementation and performance evaluation of ICN through a testbed-based study (Nasir and Jeong 2020), highlighting improvements in content delivery time and network management carried out. Another paper (Hurali and Patil 2022) provides a comprehensive overview of the state-of-the-art applications and challenges in the field of ICN.

ICN is also integrated into another popular concept, IoT. One effort (Djama, Djamaa, and Senouci 2020) elaborates on how Information-Centric Networking solutions are particularly suitable for the Internet of Things (IoT). The shift from a location-centric to a data-centric communication paradigm allows for more efficient handling of Named Data Objects (NDOs) through mechanisms like name resolution and direct naming. This approach can significantly enhance the performance and scalability of IoT systems. There are also some review/survey studies (Din, Asmat, and Guizani 2019; Mars et al. 2019) that discuss the adoption of ICN in IoT, motivated by ICN's advantages like content caching and the decoupling of senders and receivers, which are beneficial for IoT's distributed and dynamic nature. There's another study (Gündoğan et al. 2020) that focuses on presenting the ICN as a promising approach for the Industrial Internet of Things (IIoT), particularly for exchanging content chunks with sensors and actuators that may be intermittently connected or mobile.

2.2. Named Data Networking of Things

Among the numerous proposals for ICN, the framework developed within the NDN (Named Data Networking) Project (Piro et al. 2014; Zhang et al. 2010) project elicited substantial interest from many communities, including the academic community. At its core, NDN uses unique names to identify data instead of using addresses to identify locations. This means that users request data by naming what they want rather than specifying where to send packets.

In the first stages, (Jacobson et al. 2009) introduced CCN (Content Centric Networking), which treats content as a primitive, decoupling location from identity, security, access, and enabling content retrieval by name. This shift aims to resolve issues like unreliable content accessibility

and complex content-host mapping, promising enhanced scalability, security, and performance. CCN's reimagined approach reflects a move towards an internet architecture better suited to the modern era's content-driven demands. In the quest for more effective and efficient mobile networks, particularly ad-hoc networks, NDN emerged as a transformative approach. A study (Meisel, Pappas, and Zhang 2010) explores the limitations of traditional IP-based networking in mobile environments, emphasizing the challenges of infrastructure dependence and the dynamic nature of node interconnectivity. They advocate for NDN's paradigm shift, which replaces IP with named data, thus altering communication from destination-oriented to content-oriented, a move that simplifies and improves ad-hoc/mobile networking.

The application of NDN to real-world scenarios has been demonstrated by studies like PCLive (Liang et al. 2023), which brings NDN to internet livestreaming. This research not only showcases NDN's potential in handling real-time, high-bandwidth content but also addresses design issues related to protocol translation and security, further proving NDN's adaptability and efficiency. Additionally, the exploration of secure packet encapsulation within NDN networks indicates a strong focus on enhancing data security and integrity, which is crucial for the broader adoption of NDN technologies. Also, another study (Patil et al. 2023) focuses on decentralized photo sharing, highlighting applications, decentralization, and file-sharing aspects in NDN. As another example of real-world use cases, a study (Timilsina et al. 2023) created a dataset of NDN traffic traces and a toolkit for capturing, analyzing, and replaying these traces derived from real-world NDN routers.

There are studies that collectively explore the application and potential benefits of NDN and CCN within the IoT. A study (Hail 2019b) discusses NDN's potential to address IoT's diverse and resource-constrained environment, offering a promising technology to improve quality of life through various real-life applications like healthcare and smart cities. It emphasizes NDN's architecture, focusing on its ability to optimize power supply and efficiently distribute data in the network. Another study (Aboodi, Wan, and Sodhy 2019b) argues that NDN and CCN address critical IoT challenges, such as limited memory, computational power, and energy efficiency, especially in environments with unstable network connectivity. NDN is highlighted for its scalable content distribution, enhanced mobility, and robust security; all facilitated

through application-specific hierarchical naming and forwarding strategies. This approach simplifies content access and eliminates the need for IP address assignments, thus improving efficiency in extensive IoT networks. There's also a study (Gündoğan et al. 2018) that presents a comprehensive analysis comparing the performance, robustness, and efficiency of the NDN, CoAP, and MQTT. Another detailed paper (Baccelli et al. 2014a) represents a significant contribution to the domain of ICN within the IoT by documenting the innovative process of porting CCN-Lite to RIOT OS. It meticulously details the first NDN experiments in a real-world, large-scale IoT deployment, highlighting the challenges and potential of ICN in IoT. The study critically evaluates CCN's suitability for IoT, proposing and assessing enhancements that reduce control traffic and optimize data handling to meet stringent energy and bandwidth constraints. Moreover, it enhances content availability for intermittently active IoT nodes and contrasts CCN's performance with conventional IoT protocols such as 6LoWPAN/RPL/UDP, offering an unprecedented comparative analysis. This research is instrumental in advancing the understanding and implementation of ICN approaches in IoT infrastructures.

2.3. Forwarding Mechanisms in NDNoT

Forwarding mechanisms in NDNoT play a crucial role in the efficient propagation of Interest and Data packets across the network. In NDNoT, data is accessed through the address of Data packets. Interest packets are tasked with seeking out this Data. Forwarding mechanisms determine the routes for Interest packets, directing them to the locations where the Data is available. The most basic method, Interest flooding, involves broadcasting Interest packets to all neighboring nodes or across a broad area without considering the specific data source's location. This flooding-based approach aims to maximize the likelihood of finding the requested content within the network by forwarding Interests extensively. However, this method often leads to inefficient network resource utilization, increased bandwidth consumption, and potential redundancy in the retrieval process. While it may increase the probability of finding the desired content, it lacks the precision and scalability of more targeted routing strategies, potentially resulting in unnecessary network congestion and reduced overall efficiency. However, through the management of Interest floods, this method effectively directs Interest packets towards their respective Data sources. These simple blind forwarding

mechanisms (Amadeo, Campolo, and Molinaro 2013; Varvello et al. 2011; Wang et al. 2012) provide a simple solution for forwarding. Although, even these solutions might not be suitable enough for different NDNoT scenarios/topologies.

Unlike blind flooding, awareness based forwarding (such as content aware, energy aware) methods aim for targeted, informed routing decisions. Awareness based forwarding approaches intelligently direct Interest packets toward potential Data sources, minimizing unnecessary transmissions and optimizing resource utilization. An approach (Lu et al. 2013) provides two additional packet types: CMD and ACK. By interchanging these message types between one-hop neighbors, the mechanism determines the next hop. Another study (Li, Liu, and Okamura 2013) introduces Hello messages that traverse over the network to determine and optimize paths. Next hop selection is done by greedy approach while this mechanism is also QoS aware. Other than next-hop awareness, LFBL (Meisel n.d.) and E-CHANET (Amadeo, Molinaro, and Ruggeri 2013) mechanisms are based on distance/provider awareness. These approaches select the best performance path if the content is available from multiple sources. MANET CCN (Oh, Lau, and Gerla 2010) provides two types of routing: content pulling and content pushing. TOP-CCN (Kim, Shin, and Ko 2013) is a proactive approach where the provider node broadcasts the name prefixes. Upon receiving these broadcast messages, neighboring nodes add this information to their FIB tables. NAIF (Yu et al. 2013) is a neighbor-aware approach. This approach reduces overhead and mitigates multiple copies of the same Interest messages. Another neighbor-aware approach, BlooGo (Angius, Gerla, and Pau 2012), tries to minimize message transmissions over the network by using the bloom filter gossip algorithm. BOND (Meisel 2011) adds a header field that contains forwarding information to each Data packet. Every node receiving Data updates its forwarding table with this information. It's also possible to select forwarding direction with geo-aware forwarding mechanisms. BREB (Han et al. 2014) introduces PRT (Pending Request Table) and CPT (Content Provider) tables. PRT holds unsatisfied Interests, while CPT holds Providers who responded with Data. BREB focuses on the shortest path by utilizing TTL values.

In sensor networks and IoT environments with constrained hardware, energy efficiency is an important issue. REMIF (Rehman et al. 2016) is a forwarding mechanism that focuses on this

energy efficiency. In REMIF, the Interest message forwarding mechanism is dependent on the energy status of the nodes. OEFS (Rehman, Ahmed, and Kim 2017) mechanism focuses on a proactive approach while considering the energy of nodes. If the energy of a node is critical, that node does not forward incoming Interests to conserve lifetime. Instead, it focuses on satisfying PIT entries while using the OEFS mechanism.

Congestion control is another issue in communication networks. In NDN architecture, this issue can be solved by the forwarding mechanism itself. Such a study, the SIRC (Amadeo, Molinaro, et al. 2014) mechanism, is able to detect congestions and send congestion information to neighboring nodes. On another mechanism (Rozhnova and Fdida 2012), each node detects congestion and holds its own forwarding rate. On the other hand, the HR-ICP (Carofiglio, Gallo, and Muscariello 2012) mechanism keeps track of Interest and Data rates and uses the ChoPCoP (F. Zhang et al. 2014) protocol to observe and detect Data queue lengths.

One study (Touati, Aboud, and Hnich 2022) introduces a novel NDN-based communication model that integrates an energy-aware forwarding strategy with a smart sleep mode, significantly optimizing energy usage within IoT networks. Complementing this, another paper (Djama et al. 2022) proposes an adaptive forwarding strategy, LAFS, which leverages machine learning to refine decision-making in data forwarding, thereby bolstering network performance and reliability. Further research (Ngo 2022) investigates routing protocols within NDN-based ad-hoc networks, underscoring NDN's potential to streamline data dissemination and retrieval in dynamically changing network topologies. Additionally, there's a focus (Marques, Senna, and Luís 2022) on forwarding mechanisms within energy-constrained wireless information-centric networks, highlighting the critical need for energy-efficient approaches to maintain network sustainability and device longevity. Collectively, these studies underscore a trend towards more intelligent, adaptive, and energy-efficient networking solutions in the realm of IoT.

While a spectrum of specialized forwarding mechanisms exists for NDNoT, not all strategies align seamlessly with networks housing constrained IEEE 802.15.4 devices. The distinctive nature of these devices, characterized by limited system resources, requires forwarding

mechanisms that are notably lightweight and resource-efficient. Traditional forwarding strategies designed for more robust systems may not be suitable options for networks reliant on constrained devices. VIF (Vanilla Interest Flooding) and RONR (Reactive Optimistic Name-based Routing) (Baccelli et al. 2014b) mechanisms proposed for such constrained devices. VIF forwards Interest messages blindly by broadcasting them. RONR is similar to VIF at the beginning phase. However, as Data starts to flow, each node keeps track of name prefixes in their FIB. This way, they would be able to forward Interest messages to interfaces that are recorded on FIB.

Within the literature on NDNoT, numerous studies have been focused on exploring various forwarding strategies. While forwarding mechanisms generally take center stage, other integral components like caching, the sizing of PIT and FIB, and methods to mitigate Interest flooding attacks are equally pivotal aspects of the forwarding mechanism. These structures and mechanisms collectively contribute to the efficient propagation and retrieval of Interest and Data packets in NDN-based networks. Each element plays a critical role in optimizing the network's performance, ensuring robustness, scalability, and effective content delivery within the NDNoT paradigms. Especially in constrained IEEE 802.15.4 NDNoT networks, each component needs to be suitable and lightweight for these devices.

2.4. Interest Flooding Attacks and Mitigating Interest Flooding Attacks in NDN & NDNoT

Content is searched and retrieved by Interest messages in an NDN environment. However, flooding these Interest messages into an NDN may disrupt communication by causing too much network traffic. A malicious node in an NDN environment can transmit Interest messages constantly to render NDN unstable. So, the mitigation of interest flooding attacks (Afanasyev et al. 2013; Hidouri et al. 2022; Lee et al. 2022; Rai and Dhakal 2018) in NDN environments holds a critical role in ensuring the overall security and reliability of these network architectures. As far as known, there's only one study (Xue et al. 2017) on mitigating interest flooding attacks in the domain of NDNoT. However, it's important to note that this research was carried out in the NDNSim environment, which might not accurately represent the

conditions of the NDNoT environment. Most of the existing interest flooding attack mitigation techniques are intended for NDN networks. However, some of these approaches might be adapted to NDNoT.

Most IFA (Interest Flooding Attack) mitigation approaches in the literature are statistical-based. An approach (Qureshi et al. 2021) provides dynamic threshold values to detect malicious Interest packets. DMNWV (Cheng et al. 2020a) proposes a method that is able to detect and mitigate IFAs that start slowly and get faster to deplete PIT. Another study, AH-IFAC (Zhang and Li 2019), is based on AutoregRessive Integrated (ARI) and Hidden Markov Model (HMM) to mitigate IFAs. Also, based on one study (Hou et al. 2019), theil-based countermeasures can also mitigate IFAs. An attacker node can transmit non-satisfiable Interest messages to deplete PIT of nodes in the network. In such cases, the Interest dissatisfaction ratio of nodes can be used (Pu, Payne, and Brown 2019) to mitigate IFAs. It is also possible to detect and mitigate IFAs through congestion awareness (Benmoussa et al. 2019). Another approach (Wang et al. 2013) mitigates IFAs with Disabling PIT Exhaustion (DPE) by directly recording state information of malicious Interests/neighbors.

While statistical-based approaches might be sufficient in some localized scenarios, malicious Interest messages can disrupt another part of the network. There are also collusive statistical-based approaches in the literature to focus on this issue. For example, the CUSUM algorithm (Al-Share, Shatnawi, and Al-Duwairi 2022) is specialized for such scenarios. I-CIFA (Wu et al. 2021) is a study that focuses on collusive IFAs that inspect such collusive attacks. In another study (Wang et al. 2014), nodes in the network inform each other about possible attacker nodes. The Poseidon approach (Compagno et al. 2013) is also another example of a study on this scope.

In an NDN environment, it is also possible to use machine learning based approaches to mitigate IFA. Such methods leverage the power of AI and data-driven algorithms. One example study, GNN4IFA (Agiollo et al. 2023), detects IFAs utilizing graph neural networks. Another study, MF-RF (Yue, Peng, and Feng 2023), is based on multiple features with random forest. This approach mitigates improved collusive attacks mentioned in the previous paragraph. In addition, it is also possible to use the Long Short-Term Memory (LTSM) method (Zhang, Li,

and Hou 2022) to detect and mitigate IFAs. Through feature selection (Kumar, Singh, and Srivastava 2021), the network can use these features to detect and mitigate such IFAs. In (Chen et al. 2019), isolation forest is used to detect IFAs. Using entropy-svm and jensen-shannon divergence is also possible, according to another study (Zhi et al. 2020). As there are many machine learning based methods that can be used for detecting and mitigating IFAs, a study (Kumar, Singh, and Srivastava 2017) evaluates machine learning algorithms for this purpose.

Another approach to mitigate IFAs is securing the Interest messages. Cryptography-based methods employ encryption and digital signatures to protect Interest and Data packets. The PERSIA (Tourani, Torres, and Misra 2020) acts as a defense mechanism against such attacks by introducing a puzzle-based approach. It incorporates computational puzzles into the interest packets sent by the users. These puzzles require the requester to solve a computational challenge (such as solving a cryptographic puzzle or performing a CPU-intensive task) before the Interest packet is forwarded by the network nodes. Another study (Alston and Refaei 2016) involved implementing cryptographic tokens within the routing process of NDN to counter IFAs. By embedding these tokens in the routing information, the network nodes verify the authenticity and integrity of the routing updates. This process ensures that only authorized nodes with valid cryptographic tokens can propagate routing information. Consequently, it prevents malicious entities from flooding the network with excessive interest requests, as only authenticated nodes can participate in the routing process, thus enhancing the network's resilience against IFAs in NDN environments. Interest Cash (Li and Bi 2014) introduces a caching mechanism at the application layer, specifically designed to mitigate IFAs targeting dynamic content retrieval in NDN. This approach involves caching previously satisfied content requests at the application level, allowing subsequent requests for the same content to be fulfilled locally without generating additional Interest packets. Intelligently, caching and managing dynamic content requests reduces the network's susceptibility to interest flooding, minimizes unnecessary Interest packet propagation, and enhances overall network efficiency and resilience in handling dynamic content retrieval in NDN environments.

The NDN architecture relies significantly on its forwarding strategy, an essential element in managing the propagation of interest packets within the network. Through strategic and attack

aware forwarding mechanisms (Araujo, Madureira, and Sampaio 2023), NDN networks effectively handle challenges posed by IFAs, prioritizing legitimate requests while identifying and addressing malicious or irrelevant ones. Implementing these strategies involves intricate algorithms and decision-making processes, demanding a careful balance in allocating resources, especially in resource-limited IoT settings. Despite these complexities, forwarding strategies remain a milestone in NDN architecture and maintain the network's defense against both individual and collaborative IFAs.

2.5. Caching in NDNoT

In NDNoT, caching plays a pivotal role in optimizing data retrieval and network efficiency, just like in NDN. NDNoT's caching mechanism strategically stores data along the network path, enabling subsequent requests for the same content to be fulfilled locally rather than fetching it from the original source. This caching strategy significantly reduces redundant data transmission and alleviates network congestion by leveraging the principle of content-centricity. As Interest packets traverse the network and encounter content, NDNoT routers or forwarders intelligently cache Data packets associated with these Interests. Cached data stays on router memory, forming a temporary repository based on popularity, allowing faster responses to future requests. The caching mechanism enhances network responsiveness, reduces latency, and promotes efficient data distribution across the NDN architecture. However, efficient cache management and storage optimization remains an ongoing area of research within the NDNoT framework. Although there are various studies (Herouala et al. 2023; Kim et al. 2016; Muto, Kanai, and Katto 2015; Yu et al. 2017) on NDN caching, these mechanisms may not be suitable for more diverse NDNoT environments.

A study (Amadeo et al. 2020) focuses on optimizing IoT content distribution by implementing caching strategies at the network's edge within the context of NDNoT. This research focuses on efficiently storing frequently accessed and updated data at the edge and takes advantage of NDNoT's content-centric approach. By strategically caching popular and fresh IoT content nearer to end-users, this study aims to minimize latency, reduce bandwidth usage, and enhance overall network performance. Another approach (Meddeb et al. 2019) focuses on refining

caching methodologies by prioritizing the removal of the least fresh content from the cache memory. By adopting this policy, the study aims to optimize data storage and retrieval efficiency in NDNoT networks. This cache replacement approach aligns with the dynamic nature of IoT environments, ensuring the retention of more relevant and updated data in the cache. Furthermore, another research (Alahmri, Al-Ahmadi, and Belghith 2021) aims to optimize data retrieval and storage mechanisms by implementing collaborative cache management strategies and resource pooling methodologies. By encouraging collaborative caching mechanisms and resource pooling, this approach aims to optimize data management processes and enhance the overall efficiency of NDNoT. Additionally, within NDNoT, there are hybrid caching management approaches that blend multiple strategies for improved efficiency and data accessibility. One of these studies (Naeem et al. 2022) is a hybrid approach that is able to do content selection, content placement, and content replacement. Another study (Gupta et al. 2020) focuses on multiple caching mechanisms/policies in the domain of NDNoT.

2.6. PIT/FIB Sizing for NDNoT and NDN

Central to NDN's efficient operation is the Forwarding Information Base (FIB), which is used for routing and forwarding Data based on content names, and Pending Interest Table (PIT), which temporarily stores Interest information until the requested Data is returned. Correctly sizing the FIB and PIT is crucial for balancing resource utilization with network performance. This chapter explores various FIB and PIT sizing and optimization strategies within NDN, offering insights into their methodologies, results, and potential impacts on network performance and scalability. Currently, in the literature, FIB/PIT size studies conducted specifically on NDNoT are limited. However, FIB/PIT size studies performed on NDN might be compatible with NDNoT environments.

A study (Mun and Lim 2019) explores the concept of shared FIB tables in NDN. The authors propose a novel approach where multiple routers can share a single FIB table, aiming to reduce memory overhead and enhance scalability. The paper delves into algorithmic strategies for efficient FIB sharing and evaluates the trade-offs between resource utilization and routing

performance. The critique of this method reveals its potential in large-scale networks while noting the complexities involved in table management and update synchronization.

BFAST (Dai et al. 2015) addresses the challenges of FIB scalability by introducing a unified and scalable index mechanism tailored for NDN's forwarding architecture. The paper outlines the design principles of BFAST, emphasizing its adaptability to various network sizes and traffic patterns. The proposed structure is assessed through extensive simulations, showcasing its ability to maintain high throughput and low latency. The review highlights BFAST's innovative indexing technique but also discusses limitations in terms of initial setup complexity and maintenance.

FCTrees (Karrakchou, Samaan, and Karmouch 2020) contribute to the discourse on FIB optimization by presenting a compression method that reduces the memory footprint of FIB structures. The paper introduces a family of tree-based FIB structures employing front-coding techniques to compact common prefixes and minimize space requirements. The effectiveness of FCTrees is measured in scenarios with varying prefix distributions, demonstrating significant savings in memory. The critique acknowledges the method's space efficiency but also considers the computational overhead associated with compression and decompression processes.

One comprehensive review (Rosa and de Oliveira Silva 2022) surveys recent advancements in FIB implementations specifically designed for high-speed switches in NDN. It covers a range of techniques, from trie-based structures to hardware-assisted solutions, providing a comparative analysis of their performance metrics, such as lookup speed, memory efficiency, and update dynamics. The paper offers valuable insights into the suitability of each method for different network environments and the trade-offs involved in achieving speed and scalability.

On the subject of sizing the PIT, a study (Carofiglio et al. 2015a) addresses the critical challenge of determining the optimal size for the PIT in NDN. The authors propose a model that considers various network parameters, such as traffic volume and content popularity, to dynamically adjust PIT size. The study presents a detailed analysis of how PIT sizing affects network throughput and latency. It suggests that a well-dimensioned PIT can significantly enhance

network performance while minimizing resource waste. The critique focuses on the model's adaptability to real-world network conditions and its scalability in larger, more complex environments.

Moving from theory to practical implementation, another paper (Yuan and Crowley 2014) explores scalable designs for the PIT that can adapt to growing network demands. It outlines several architectural principles essential for a scalable PIT, including efficient data structures, hash-based indexing, and strategies for handling interest collisions. The authors provide a comparative analysis of different design choices and their impacts on lookup speed, memory efficiency, and overall system scalability. The review appreciates the thoroughness of the comparative study but also discusses the practical challenges of implementing these designs in diverse NDN scenarios.

Another study (Dai et al. 2012) offers a deep dive into the operational intricacies of the PIT in NDN. It examines how the PIT's performance and efficiency are influenced by various factors like interest aggregation, timeout policies, and network topology. The paper also discusses the security implications of PIT sizing, highlighting how an improperly sized PIT can be vulnerable to certain types of attacks. The critique acknowledges the comprehensive analysis of operational aspects but suggests that further research is needed to integrate these considerations into a PIT sizing strategy.

Focusing on the challenges brought by mobility in Information-Centric Networks (ICN), which encompass NDN, a study (Sivaraman, Guha, and Sikdar 2020) investigates how mobile producers affect the optimal size of the PIT. It introduces a dynamic model that accounts for the mobility patterns of producers and the resulting changes in data availability and network paths. The study's simulations demonstrate the model's effectiveness in various mobility scenarios, suggesting that a dynamic, context-aware PIT sizing approach is necessary for mobile environments. The critique examines the assumptions made about mobility patterns and the potential complexities of implementing such a dynamic model in a real-world setting.

3. MATERIALS AND METHODS

This study is an applied research focusing on an NDNoT protocol stack that can work on constrained IEEE 802.15.4 devices. The study, which started in August 2021 at Adana Alparslan Türkeş Science and Technology University, was completed in December 2023. In this section, the materials used in the study and the methods applied are discussed in detail.

3.1. Materials

This section contains a comprehensive overview of the materials utilized throughout this study. It covers the various approaches and techniques chosen and applied to conduct the research, providing a clear roadmap of the systematic processes used to gather.

3.1.1. Contiki NG OS

Contiki NG OS stands out as an ideal operating system for constrained IEEE 802.15.4 IoT Networks due to its compatibility with various essential protocols. It seamlessly supports crucial protocols such as IPv6/6LoWPAN, CoAP, and MQTT, among others, which are essential for efficient IoT communication. Its adaptability to these critical protocols makes it a solid platform for IoT network deployment. Moreover, Contiki NG OS is known for its straightforward architecture, simplifying the creation of novel protocols and applications with considerable ease. This characteristic ease of development makes it an attractive choice for researchers and developers seeking to innovate and develop protocols for specific IoT applications. Contiki NG OS stands as an open-source platform and has a large community that offers comprehensive support, well-documented resources, and collaborative contributions. Contiki NG OS also provides its fully compatible simulation tool, Cooja Network Simulator.

The core of Contiki OS is primarily composed of multiple lightweight event schedulers and a polling mechanism. Events executing processes with the help of the event scheduler, which also

periodically invokes the polling handlers of processes. The actions of the polled process are distinguished by these handlers. Conversely, the polling mechanism identifies high-priority events and is utilized by processes operating closely with hardware to monitor the status updates of hardware devices. Processes with poll handlers are called upon in accordance with their priority. It encompasses services for handling sensor data, communication protocols, and device drivers. Each service is equipped with its own interface and implementation.

Contiki's programming models facilitate both multithreading and event-driven approaches through the utilization of protothreads. Protothreads offer a significant benefit in their exceptionally low memory usage, requiring no additional stack for a thread. Handlers in Contiki are prevented from being interrupted to initiate new events as events progress to completion. Contiki OS does not allow process synchronization.

3.1.2. Simulation Environment – Cooja Network Simulator

Cooja Network Simulator (Figure 3.1.1) is a powerful network simulator specialized for wireless sensor networks and IoT (and as a result of this study, NDNoT) devices, and it's an integral component of the Contiki NG OS. This simulator provides a simulated environment where users can replicate diverse network topologies and evaluate the behavior of applications and protocols within these virtual setups. Using Cooja's capabilities, developers and researchers can conduct comprehensive testing, assessing the performance and reliability of their protocols and applications in controlled settings. With its ability to mimic real-world scenarios, Cooja facilitates the analysis of network behavior, aiding in the optimization and refinement of protocols designed for IoT environments. Seamlessly integrated with Contiki NG OS, Cooja extends support for developing, testing and validating IoT protocols and applications, contributing significantly to the robustness and efficiency of IoT networks. In the simulation environment, development and performance analysis can be done quickly. The effect of changes/developments can be achieved almost instantaneously. For these reasons, the simulation environment was primarily preferred in the study.

The cooja network simulator effectively emulates Skymote and Zolertia Z1 IEEE 802.15.4 devices. However, due to Skymote's limited system resources, Zolertia Z1 was selected as the preferred simulated device for the study. This choice was motivated by the Z1's compatibility with the desired system requirements, ensuring a more accurate representation of constrained IoT devices within the simulation environment.

In the performance analyses carried out on the simulation, different simulation variables were used in order to see the effects of the used / selected protocol more clearly. The different simulation variables used are detailed under the relevant method section of the protocols.

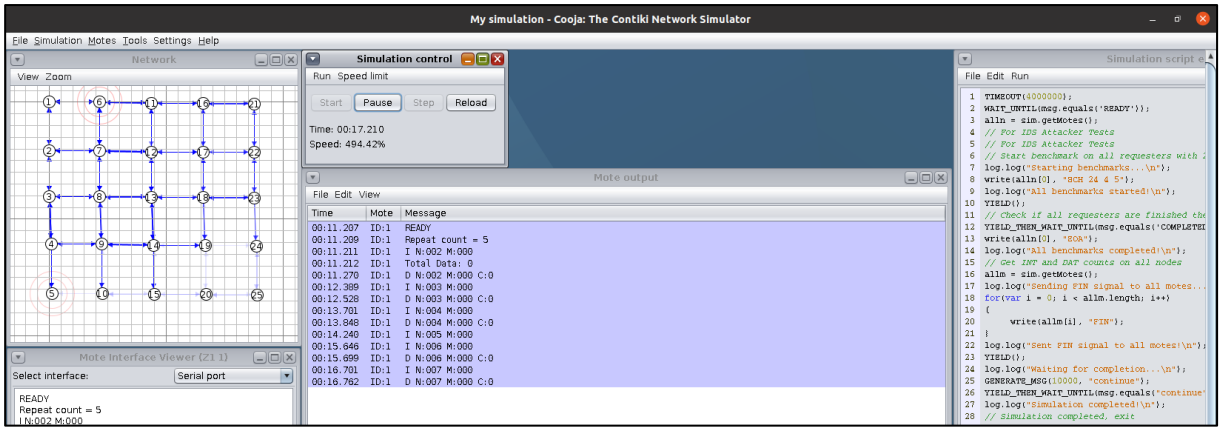


Figure 3.1.1. Cooja Network Simulator with Sample Network and Sample Simulation Script on Ubuntu Linux

3.1.3. Hardware (Real-world) Environment – OpenMote IEEE 802.15.4 IoT/NDNoT Device

The OpenMote series is a set of open-source IoT/NDNoT hardware platforms specifically designed to facilitate research, experimentation, and development tasks. These devices are engineered with wireless communication functionalities, offering a range of features, including sensors, connectivity options, and energy-efficient operations. Valued for their adaptability and accessibility, OpenMote devices serve as essential tools for exploring diverse IoT concepts, implementing various protocols, and conducting experiments across a spectrum of IoT

environments. Widely employed in academic research, IoT development, and prototyping, the OpenMote series remains a go-to choice due to its versatility and compatibility with a broad array of IoT applications and scenarios. OpenMote devices utilize the 802.15.4 radio standard for their wireless communication, enabling low-power and low-data-rate connectivity in IoT networks. While developments and performance analyses were conducted in a simulation environment, real-world operability was also tested on OpenMote devices for accurate results. While having a CC2538 chipset, the OpenMote ecosystem allows various shields such as USB shield, battery shield, and OpenBase.

Figure 3.1.2. OpenMote IEEE 802.15.4 Devices, from Left to Right; OpenMote with USB shield, OpenMote with OpenBattery, OpenMote with OpenBase

3.2. Methods

3.2.1. µNDN Protocol Stack

The Internet's rapid growth has led to incredible advancements in home systems, healthcare, city development, and industry automation. These breakthroughs rely on smart devices, limited yet clever tools that sense, gather, and respond to data around them. This trend, known as the Internet of Things (IoT), marks a major tech wave powering various life-changing applications in our world today. IoT applications are based on services centered around content. However, the traditional Internet design is centered around hosts. This difference has prompted extensive exploration into the development of Future Internet Architectures (FIAs). Among these, Named Data Networking (NDN) stands out as a promising FIA, perfectly suitable for the communication demands of IoT networks. Additionally, due to its independence from heavy-weighted protocols like IPv6, 6LoWPAN, TCP, etc., the NDN architecture is better suited for constrained IoT environments. As a result, NDN architecture and protocols are considered to be implemented for IoT devices. Despite the limited existing implementations, there appears to be a notable absence of an open-source NDN implementation specifically designed for constrained IEEE 802.15.4 networks/devices, as far as current knowledge suggests.

The use of NDN architecture in limited IoT networks (NDNoT) is an emerging area of study. Although it is still being worked on, there are many issues that can be studied. Taking these into account, this study focuses on the development of the µNDN protocol stack, aiming to facilitate future research and developments in this domain.

3.2.1.1. Constraints & Challenges

IoT devices, specifically the IEEE 802.15.4 IoT devices, typically have highly constrained hardware. Consequently, protocols and applications designed for these devices are either optimized versions of existing ones or custom-built solutions developed specifically to function within these constraints. In comparison to the 6LoWPAN/IPv6 stack, the NDN architecture

emerges as a smoother protocol stack. However, limitations persist within the NDN architecture, particularly when applied to 802.15.4 IoT devices.

One significant restriction lies in the packet size. The maximum transmission unit (MTU) for IEEE 802.15.4 IoT networks is restricted to 127 bytes, including a 25-byte MAC header. Consequently, each NDN packet is constrained to carry a maximum of 102 bytes of data. While various compression methodologies can augment the payload within this limit, compliance with the 802.15.4 standard sets an upper boundary on data transmission. When considering the inclusion of 6LoWPAN/IPv6 and UDP headers, the available space for data diminishes to approximately 51 bytes. However, in contrast to this stack, NDN operates with a different header and addressing system, resulting in a larger usable field for data. Despite this allocation, certain use cases might require a larger data field. In such instances, compressing the name along with Type-Length-Value encoding in the header can be implemented. This strategy aims to expand the usable field for data, potentially accommodating larger data requirements.

In the domain of NDN architecture, structures like the Pending Interest Table (PIT) and Forwarding Information Base (FIB) are crucial components frequently referenced. Consequently, it's advantageous to store these structures in RAM memory, allowing for quicker access. However, within the constrained bounds of 802.15.4 IoT devices, RAM memory, like other system resources, is limited. Therefore, these structures need to comply with specific limitations to ensure optimal functionality without exhausting available memory.

Moreover, caching, like FIB and PIT structures, is prioritized to reside in RAM memory for swift access. However, there's an alternative option of storing cache data in flash memory, which, although limited, offers more storage space compared to RAM.

Another critical aspect is naming within the NDN architecture. The data's addressing space relies on names, which are integral components of Interest and data packages. Considering the constraints on packet size within 802.15.4 devices, as discussed earlier, it becomes crucial to maintain specialized naming schemes. The length of names directly influences

routing/forwarding mechanisms, emphasizing the necessity to determine an optimal naming field space and balancing the need for clarity with the limitations of the device's packet size.

3.2.1.2. Implementation Details

The study presents the μNDN protocol stack, an implementation of NDN for constrained IoT networks within this study's focus. Developed on Contiki NG, the latest iteration of the Contiki operating system, well-known for its popularity in 802.15.4 IoT networks. Extensive testing during the development phase was carried out in the Cooja simulation environment. While implementing the μNDN architecture, it was inspired by the Named Data Networking project. Models from this project underwent modifications and customization for IoT devices with restricted resources, effectively aligning them with the μNDN framework. Based on the NDN project architecture, the following models/structures have been implemented:

- **Header:** The μNDN utilizes a 6-byte header structure, common to both Interest and data packets, as shown in Figure 3.2.1. Investigating the possible compression of this header structure using Type-Length-Value Encoding is another route for future exploration and development. The current version of the header includes a 4-byte version, 4-byte type, 5-byte ttl, 1-byte must be fresh flag, reserved 2-byte, 16-byte nonce value, 10-byte lifetime, and reserved 6-byte.

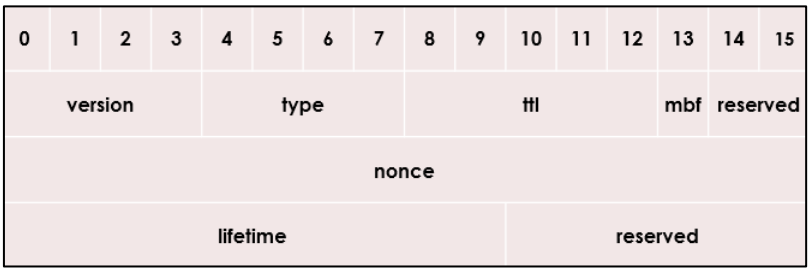


Figure 3.2.1. Shared Header Structure for Both Interest and Data Packets on μNDN Protocol Stack

• **Naming:** The µNDN operates on a 24-character naming structure comprising three segments separated by "/". The initial segment contains the node ID (content creator), the subsequent segment denotes the data category (content category), and the final segment represents the specific name of the data (content). An example of a naming scheme is given in Figure 3.2.2.

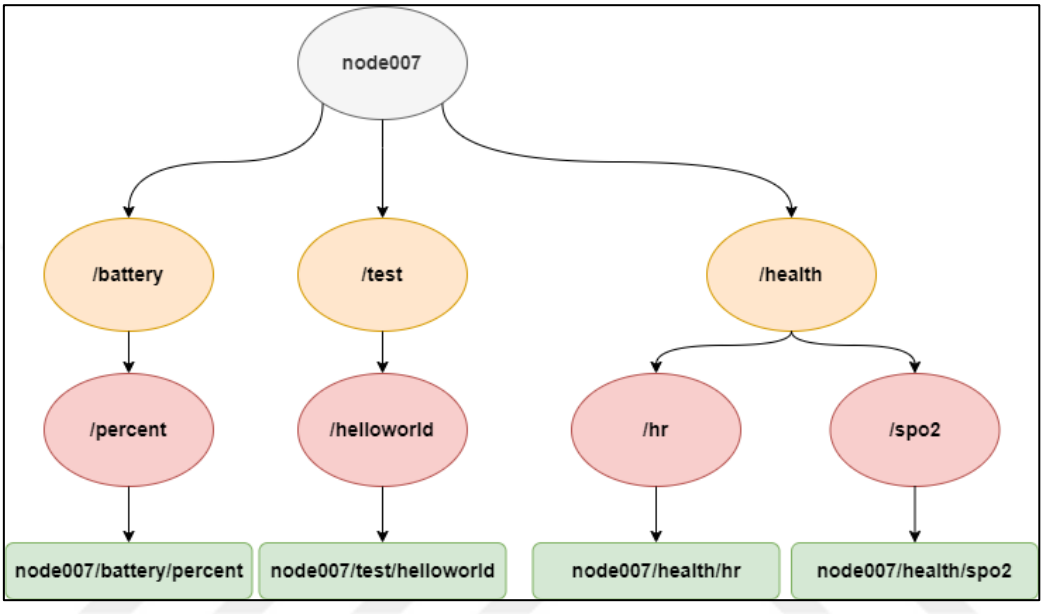


Figure 3.2.2. Hierarchical Naming Scheme that Categorizes Data Adopted on µNDN Protocol Stack

• **Interest Packet:** The Interest packet (Figure 3.2.3) is formed by a 6-byte header along with a 24-byte name. With the addition of a 25-byte MAC header, each Interest packet is 55-byte in total.

• **Data Packet:** The data packet (Figure 3.2.3) maintains a parallel structure to the Interest packet, featuring a 6-byte header and a 24-byte name. Furthermore, it allocates a 72-byte space specifically designated for the data payload. This allocation of 102 bytes (72 for data, 24 for name, and 6 for header) maximizes the MTU for the 802.15.4 network, accounting for a 25-byte MAC header.

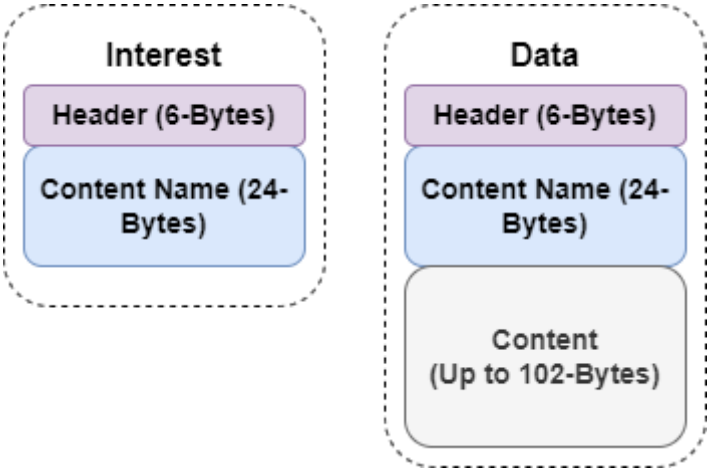


Figure 3.2.3. Interest and Data Packets, sharing 6-bytes Header and 24-bytes Name Structures on µNDN Protocol Stack

µNDN was developed with a deep understanding of the limitations present in IoT devices while staying true to the foundational principles of the NDN architecture established by the Named Data Networking Project (Zhang et al. 2010). Within the µNDN framework, each piece of data is uniquely identified by a name, and these named data units are requested and retrieved using Interest packets, staying true to the fundamental workings of the NDN architecture.

The µNDN operates on top of the 802.15.4 protocol. µNDN packets are framed by the 802.15.4. Even though µNDN is developed on top of 802.15.4, it's also feasible to utilize other link-layer protocols at the lower layer. The µNDN protocol stack operates under the assumption that no packet size exceeds the MTU of 802.15.4, which is 127 bytes. Similar to the 6LoWPAN stack, handling fragmentation in µNDN could potentially be managed by an adaptation layer as a future work.

Within the NDN architecture, forwarding mechanisms assist in the retrieval of Data packets corresponding to Interest packets. Among these mechanisms, VIF and RONR (Baccelli et al. 2014c) are notable examples, both of which have been seamlessly integrated into the µNDN system in this study. Both mechanisms are detailed in the following paragraphs.

The µNDN protocol stack is split into two internal parts: µNDN Core and µNDN Forwarder. This division aims to simplify complexity and pave the way for future improvements. The µNDN Core manages essential NDN functions and crucial databases like the Pending Interest Table, Forwarding Information Base, and Content Store. Meanwhile, the µNDN Forwarder oversees the Forwarding Mechanism(s) as its primary responsibility. A comparison between the layers of 6LoWPAN and µNDN, referencing the TCP/IP Model, is presented in Figure 3.2.4. Notably, in the µNDN stack, excluding IPv6 and 6LoWPaN reduces the system's workload significantly.

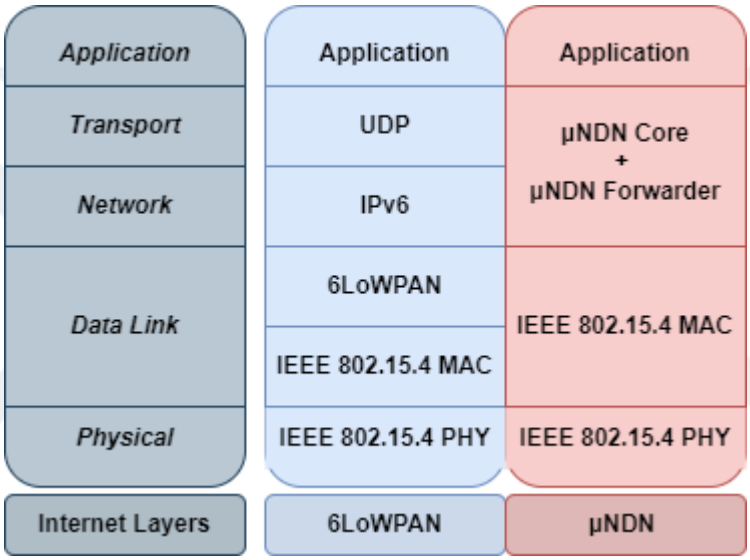


Figure 3.2.4. Traditional TCP/IP Model, Generic 6LoWPAN Protocol Stack, µNDN Protocol Stack

3.2.1.2.1. µNDN Core

The µNDN protocol implementation, similar to other NDN implementations, operates by handling Interest and data packets. Situated between the application layer and the µNDN Forwarder sub-layer, the upper µNDN Core sub-layer undertakes responsibility for fundamental NDN operations. This layer manages crucial elements such as the PIT (Pending Interest Table), FIB (Forwarding Information Base), CS (Content Store), and caching

structures, managing their functions and controls. These databases (PIT, FIB, CS) play essential roles in the functionality of the μNDN protocol, detailed below.

- **PIT (Pending Interest Table):** In the NDN architecture, data retrieval on the network is initiated through Interest messages. If a node that receives an Interest message does not have the relevant Data, it forwards the Interest message to its neighboring nodes to locate the requested Data. During the transmission of this Interest message, the node records the neighbor that initially sent this Interest in the PIT (Pending Interest Table). This mechanism enables the subsequent delivery of the requested data to the originating node through the same route. In short, PIT is a table where received and transmitted Interest messages are kept. PIT is the primary guide for the data message to find its way back. PIT basically includes the following information: Data name, interface address from which the Interest message was received, and time information when the Interest message was received. In addition to these fields, the PIT table can also be modified according to the needs of one or more protocols.
- **FIB (Forwarding Information Base):** FIB is a table in NDN networks that keeps records of where Interest packets should be forwarded based on their Data names or name prefixes. FIB tables can be populated by different methods by different forwarding mechanisms. In fact, some forwarding mechanisms (e.g., VIF) do not use the FIB table. However, this may lead to an inefficient use of the NDN network. FIB contains name prefixes, interface addresses, and registration time fields. The structure of the FIB table can be updated according to the methods to be used.
- **CS (Content Store):** The Content Store (CS) functions as the repository housing a node's data. The information stored in this area is primarily generated or updated by the application layer. Data within the CS can exist either in a static form or dynamically generated by the application layer in response to received Interest packets. Additionally, depending on the CS occupancy rate and caching mechanism, cached Data is also stored on the CS. Data on the CS is stored by matching it with data names. Additionally, a flag labels whether the data is cache data or application data.

Using these databases, the μNDN Core sub-layer reacts to received Interest and Data messages in different ways. Upon receiving an Interest packet, the first verification process involves

checking the nonce field to detect any potential loop messages. In µNDN, if a loop is detected, the packet is promptly dropped. Following this, the Pending Interest Table (PIT) undergoes inspection. The PIT serves as a data structure that retains records of received Interests that haven't yet been satisfied with data. If the PIT contains an entry matching the received Interest, it's updated accordingly. This also means an Interest message with the same Data name has already been forwarded, so it should be discarded. If there's no entry, a new entry is generated. Subsequently, the Content Store (CS) is searched for data corresponding to the Interest's name. When a match is found, the associated data is sent to the addresses aligning with the PIT record. In cases where the requested data isn't present, the Interest packet is directed to other nodes capable of fulfilling the request. This process is visualized in Figure 3.2.5.

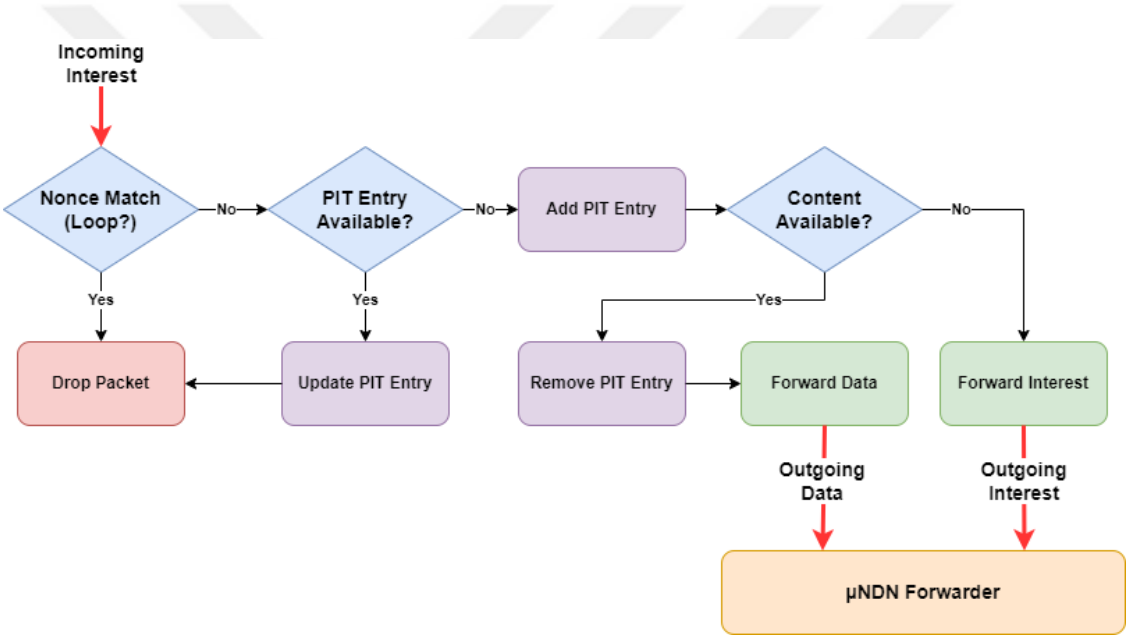


Figure 3.2.5. Reaction of µNDN Protocol Stack (for µNDN Core) for an Incoming Interest Packet

As visualized in Figure 3.2.6, upon receiving a data packet, the initial step involves a check in the Pending Interest Table (PIT). If no corresponding record exists in the PIT for the received data, the data is discarded. However, if the received data does align with an entry in the PIT, the data is transmitted to the addresses specified in the respective PIT record. Subsequently, the associated PIT record is removed.

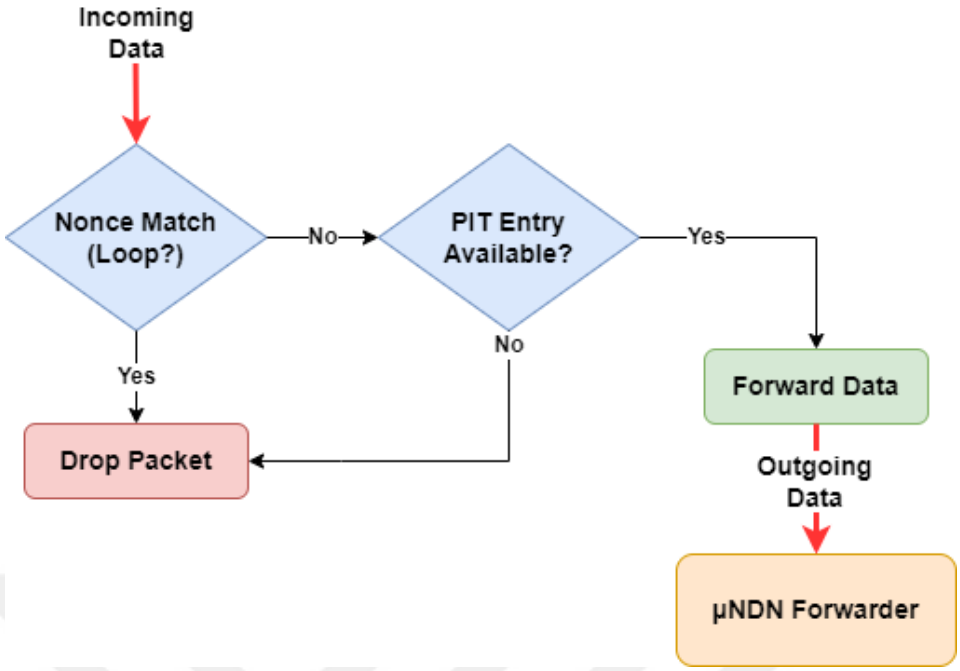


Figure 3.2.6. Reaction of µNDN Protocol Stack (for µNDN Core) for an Incoming Data Packet

3.2.1.2.2. µNDN Forwarder

The µNDN Forwarder sub-layer takes charge of data transmission in the network, organizing the forwarding of Interest and Data packets. It's the focal point for deciding the direction of Interest and data packets and determining their subsequent paths. This layer serves as the hub for implementing various forwarding strategies. The VIF and RONR protocols (Baccelli et al. 2014c) were initially integrated into this layer to assess various forwarding strategies and evaluate the functionality of the µNDN protocol stack.

- **VIF (Vanilla Interest Flooding):** VIF, shown in Figure 3.2.7, operates by forwarding an Interest packet to all neighboring nodes if the content isn't available locally. Upon receiving a Data packet, it's forwarded to nodes specified in the PIT record. VIF is resource-efficient, utilizing minimal system resources without requiring an FIB. However, it results in heavy network usage since each Interest packet behaves as a broadcast across the network.

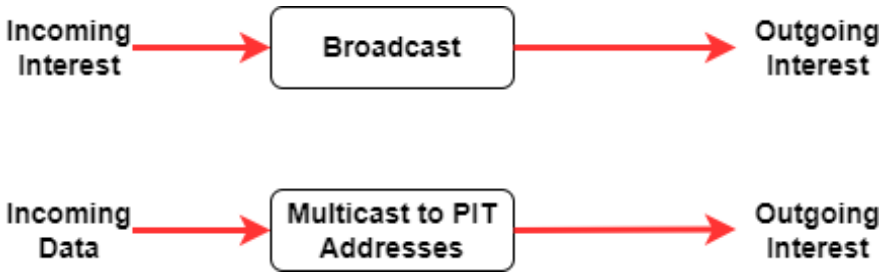


Figure 3.2.7. Reaction of VIF Forwarding Mechanism for Incoming Interest and Incoming Data Packets

• **RONR (Reactive Optimistic Name-based Routing):** RONR, shown in Figure 3.2.8, initially functions like VIF, broadcasting Interest packets across the network. When a node receives the corresponding data packet for its broadcasted Interest packet, this information is recorded in the FIB. The FIB entry contains the data name prefix and the originating node's address. For subsequent Interest packets, if their data name matches a prefix in these records, it's presumed that the Data resides at the specified address(es), and the Interest packets are forwarded accordingly.

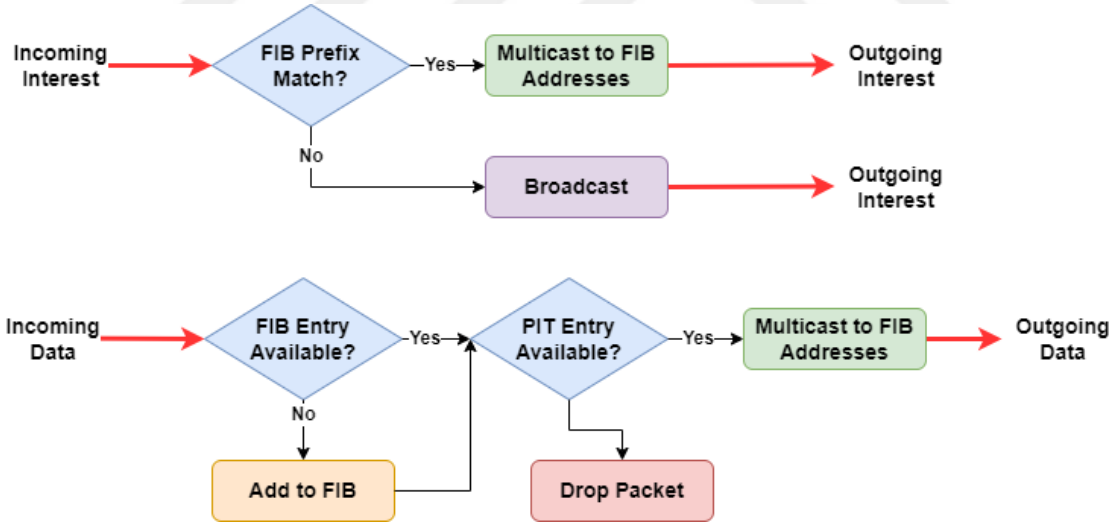


Figure 3.2.8. Reaction of RONR Forwarding Mechanism for Incoming Interest and Incoming Data Packets

The µNDN protocol stack developed within the scope of this study, and the ported VIF/RONR forwarding mechanisms were evaluated in a simulation environment. In addition, the operability of the µNDN protocol stack on real hardware, rather than relying on the simulation

environment, was tested on OpenMote IEEE 802.15.4 devices. However, since creating a testbed with a real hardware environment is both costly and troublesome, the performance analyses of the μNDN stack and VIF/RONR forwarding mechanisms, which have been proven to work in the real hardware environment, were continued in the simulation environment. The evaluations were conducted on both grid and ring topologies shown in Figure 3.2.9, each comprised of 16 nodes. To ensure accuracy, every simulation was iterated at least five times, and the outcomes were averaged. In this simulation scenario, node 1 transmits a total of 100 Interest messages to each of the remaining 15 nodes (A total of 1500 Interest messages). The simulation variables are detailed in Table 3.2.1.

Table 3.2.1. Simulation Variables used for Initial Analyses of μNDN Protocol Stack in Cooja Network Simulator

Variable	Value
Packet (Interest) Interval	200 ms.
PIT Size	30
FIB Size	30
PIT Timeout	3 secs.
FIB Timeout	300 secs.
Caching	Disabled
Interest Packet Size	55-bytes (excluding MAC header)
Data Packet Size	102-bytes (excluding MAC header)
RONR Prefix Length	7
Node Count	16
Requester Node ID	1
Total Interest Packets	15 * 100

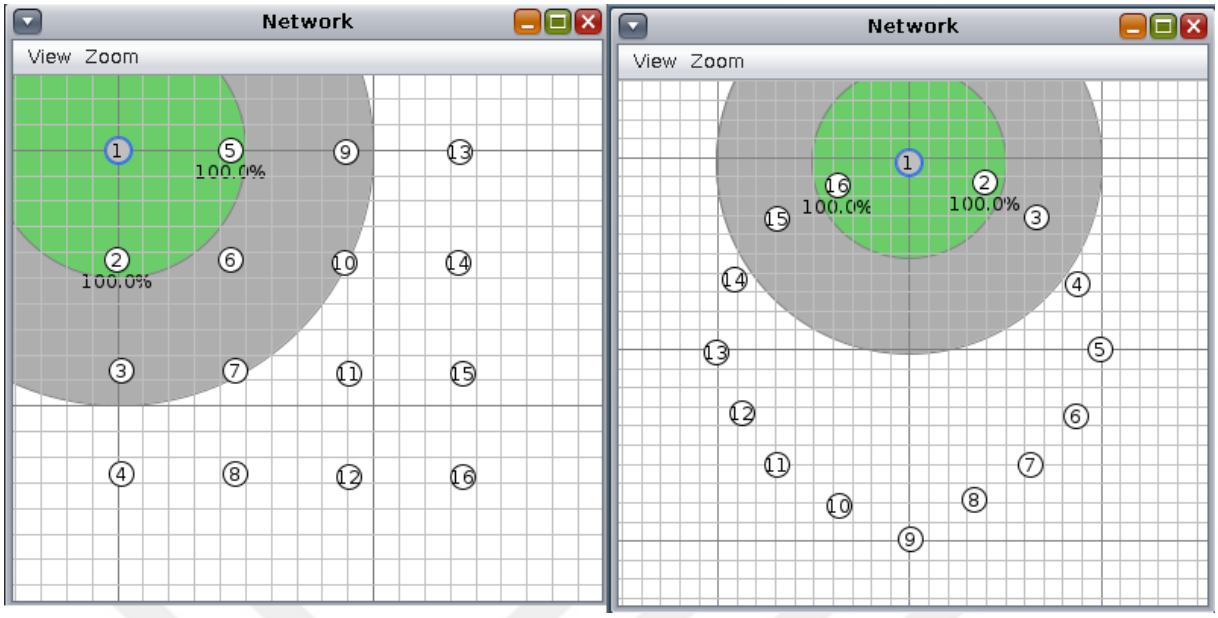


Figure 3.2.9. Grid and Ring Topologies with 16-nodes for Initial Analyses on Cooja Network Simulator

The analyses performed were examined under three metrics: Success Rate, Average Delay, and Total Network Traffic. In addition, the memory consumption of VIF and RONR forwarding mechanisms were also examined. Results for these initial analyses, specifically comparing VIF and RONR forwarding mechanisms, are outlined within the "Initial Analyses" subsection found in the "Results and Discussions" section.

3.2.2. IfNoT: An Interest Flooding Aware Forwarding Mechanism

Within an NDNoT environment, an attacker node might employ a strategy of sending Interest packets with randomly generated names that lack corresponding Data. Despite the absence of such Data packets, other nodes within the network tirelessly pursue these non-existent Data packets associated with the Interest requests. This continuous flood of counterfeit Interest packets from the attacker node leads to a substantial increase in network traffic. Consequently, this surge negatively affects communication between nodes, potentially rendering the network unusable. To counter this issue, the IfNoT mechanism was introduced specifically to detect and

mitigate these spurious Interest packets. The algorithm detailing the IfNoT mechanism is outlined in Figure 3.2.10.

Within the NDN framework, numerous attack patterns (Yue, Li, and Pang 2021) can manifest, especially those associated with interest flooding. The primary type is the continuous attack (Compagno et al. 2013; Dai et al. 2014; Salah, Wulfheide, and Strufe 2015; Wang et al. 2013). In this case, the attacker node continously sends interest packets, aiming to deplete network resources. Such continuous attack patterns can vary in intensity: they may be high-rate (Compagno et al. 2013; Dai et al. 2014; Salah et al. 2015; Wang et al. 2013), low-rate (Xin et al. 2016; Zhao et al. 2019), or even adopt an escalating rate (Cheng et al. 2020b).

In another pattern type, the attacking node may intermittently dispatch (Liu et al. 2020; Umeda et al. 2015; Xin et al. 2017) malicious interest packets. An attacker utilizing this pattern dispatches malicious interest packets for a designated period. Subsequently, the attacker ceases activity for a certain duration. This tactic is intended to evade detection by the network.

The IfNoT mechanism operates with a node table that catalogs neighboring nodes, and each is assigned an Alpha (α) value representing the probability of interest forwarding. Upon detection, a new neighbor initializes with an initial α value of 100. This α value signifies the probability percentage of forwarding an Interest packet from the respective neighbor. In a scenario with an initial α value of 100 for each neighbor, all received Interest packets are forwarded, assuming no other mechanisms affect the forwarding probability.

```
1: while Node running do
2:     if Detected a new neighbor "i" then
3:         Add i to NeighborTable
4:         NeighborTable[i][α] ← 100                          ▷ α Value set as 100 initially
5:     end if
6:     if Received Interest packet from node "j" then
7:         if Node has corresponding Data then
8:             Send Data to j return
9:         else
10:            if Entry with same name available in PIT then
11:                Update PIT time and requester
12:                Discard Interest return
13:            else
14:                Create PIT Record
15:            end if
16:            RandomNumber ← Random[0 − 100]
17:            if RandomNumber ≤ NeighborTable[j][α] then
18:                Forward Interest return
19:            else
20:                Discard Interest return
21:            end if
22:        end if
23:    end if
24:    if PIT timeout occurred for Interest from node "k" then
25:        Decrease NeighborTable[k][α]                ▷ Decrease with IfNoT Decrease Factor (λ)
26:        if NeighborTable[k][α] ≤ min_α then
27:            NeighborTable[k][α] ← min_α              ▷ α value cannot be lower than min_α
28:        end if
29:    end if
30:    if PIT record removed by successful Data forward to node "l" then
31:        Increase NeighborTable[l][α]                ▷ Increase with IfNoT Increase Factor (ψ)
32:        if NeighborTable[l][α] ≥ 100 then
33:            NeighborTable[l][α] ← 100                ▷ α value cannot exceed 100
34:        end if
35:    end if
36: end while
```

Figure 3.2.10. Pseudo Code of the IfNoT Algorithm

In NDNoT, information regarding a forwarded Interest packet is stored in the Pending Interest Table (PIT). Entries within the PIT table are removed under two conditions: upon receipt and forwarding of a corresponding Data packet or when a timeout occurs. Interest packets lacking corresponding Data packets will naturally expire within the PIT due to timeouts.

The IfNoT mechanism utilizes these PIT timeouts, interpreting a timed-out PIT entry as indicative of a counterfeit Interest packet. Consequently, when an Interest packet times out, the associated node's α value within the IfNoT table decreases by a specific multiplier known as

the IfNoT Decrease Factor (λ). Calculation formula is given in Figure 3.2.11. It's important to note that genuine Interest packets with corresponding Data might also encounter PIT timeouts. Therefore, the selection of IfNoT's λ must carefully consider avoiding the adverse impact on genuine Interest packets. This strategy aids in identifying nodes exhibiting a higher frequency of PIT timeouts, assumed to be potential attackers, and subsequently curtails their probability of Interest forwarding.

$\alpha = \alpha * \lambda$	IfNoT α Decrease Calculation

Figure 3.2.11. IfNoT New alpha Calculation with IfNoT Decrease Factor

Within the network, there might be non-aggressive nodes whose forwarding probabilities, also indicated by their α values, might have been reduced due to previous timeouts of Interest packets. As long as these nodes maintain a low α value, the likelihood of forwarding their Interest packets would remain low. To address this, the mechanism acknowledges another scenario leading to the removal of PIT records: when a Data packet is dispatched or received as a response to the initial Interest packet.

Upon the forwarding of a Data packet in response to an Interest packet, the corresponding node's α value increased by multiplication with a factor named the IfNoT Increase Factor (ψ). Calculation formula is given in Figure 3.2.12. Consequently, nodes identified as non-aggressive entities can gradually elevate their probability of forwarding Interest packets as their α values are incrementally increased.

$\alpha = \alpha * \psi$	IfNoT α Increase Calculation

Figure 3.2.12. IfNoT New alpha Calculation with IfNoT Increase Factor

Regulating the forwarding probability value (α) in the IfNoT mechanism closely resembles the process of adjusting the TCP congestion window (Paxson, Allman, and Stevens 1999). In the

case of TCP, the congestion window is fine-tuned when a timeout occurs due to packet loss, employing an Additive Increase Multiplicative Decrease (AIMD) approach. Conversely, the IfNoT mechanism adopts a different strategy, tuning the α value through a Multiplicative Increase Multiplicative Decrease (MIMD) method. Empirical evaluations have underscored the effectiveness of this distinctive approach.

During network communication, Data producer nodes might encounter disruptions for diverse reasons such as depleted batteries or radio interference. In such scenarios, authentic Interest packets seeking Data from these nodes might also time out. This situation could potentially lead to misidentifying nodes requesting Data from the affected node as attackers.

The IfNoT mechanism introduces a safeguard in the form of a minimum value, denoted as the Minimum Alpha (min_α) value, which limits the decrement of the α value to prevent such misclassifications. This ensures that the probability of Interest forwarding never reaches zero. Consequently, upon receiving Data in response to an Interest packet, the node's α value is allowed to return to its normal level. This strategy mitigates the risk of erroneously labeling nodes as attackers due to transient interruptions in communication from Data producer nodes.

Figure 3.2.13 depicts the behavior of the IfNoT mechanism within the µNDN protocol stack when it receives an Interest message. In addition, Figure 3.2.14 illustrates the behavior of the IfNoT mechanism within the µNDN protocol stack upon receiving a Data packet.

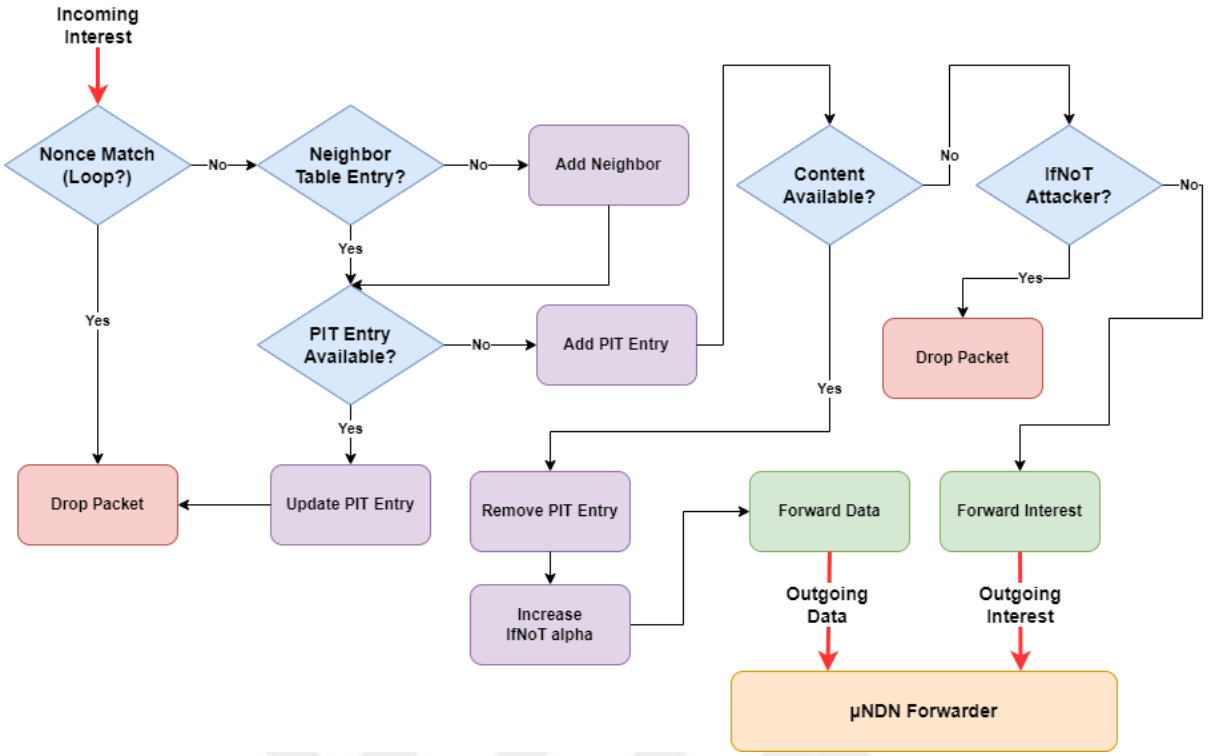


Figure 3.2.13. µNDN Incoming Interest Behaviour with IfNoT Mechanism

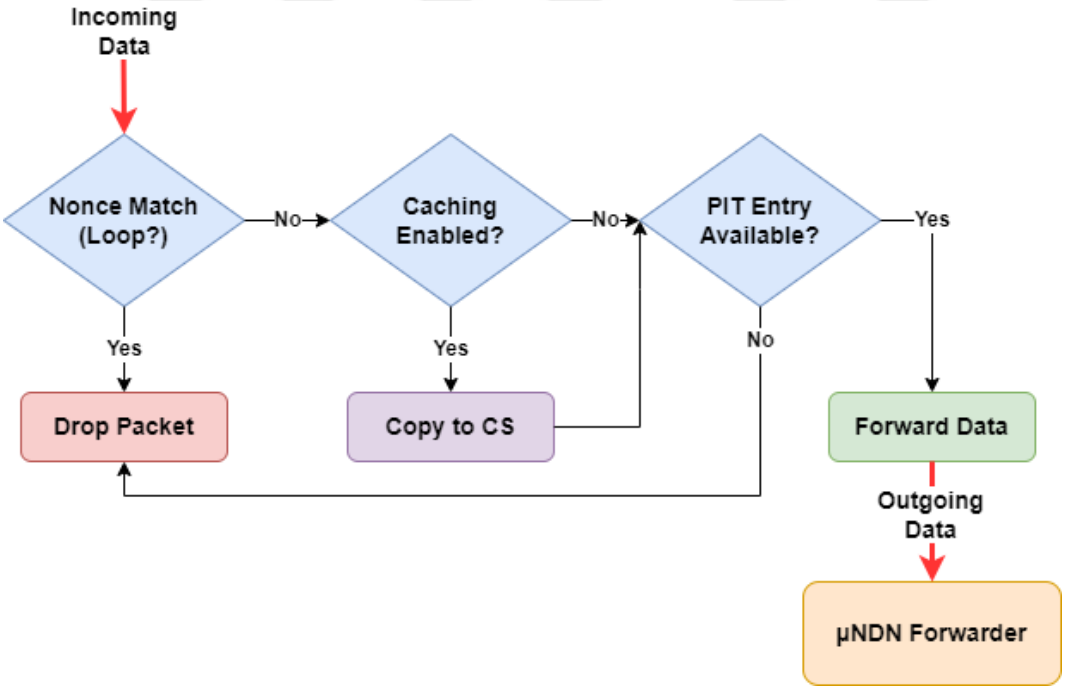


Figure 3.2.14. µNDN Incoming Data Behaviour with IfNoT Mechanism

In Figure 3.2.15, an example of topology is presented. Within this scenario, both the µNDN protocol stack and the IfNoT mechanism are active and enabled on all of the nodes. In a specific case where node-2 was suspected to be the attacker, an analysis was conducted on the neighborhood table and the corresponding α values on node-1. The progression of these values is outlined in Table 3.2.2.

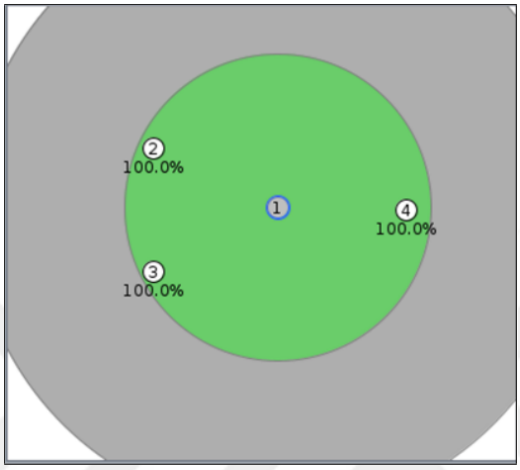


Figure 3.2.15. Sample Network with 4-nodes in Cooja Simulation Environment to Demonstrate How IfNoT Works

Due to the persistent forwarding of counterfeit Interest messages by node 2, PIT records are created on node 1. Yet, as these PIT records lack any corresponding Data, they eventually expire. As a result of these timeouts, the IfNoT mechanism decreases the α value of node-2. Even with the ongoing iterations, the α value does not drop below a specific threshold value (min_α).

Table 3.2.2. α Values for Node 1 by Iterations in Sample Network to Demonstrate How IfNoT Works

Neighbor	**α Values**			
	Initial Value	**Iteration 1**	**Iteration 2**	**Iteration 26**
2	100	90	81	10
3	100	100	100	100
4	100	100	90	100

In another instance, an Interest packet associated with node 4 expired due to the absence of its corresponding Data packet (this might happen for various reasons such as non-existent Data, radio interference, unresponsive node, etc.), leading to a decrease in the α value. However, the α value subsequently rebounded upon the receipt of Data packets in subsequent interactions.

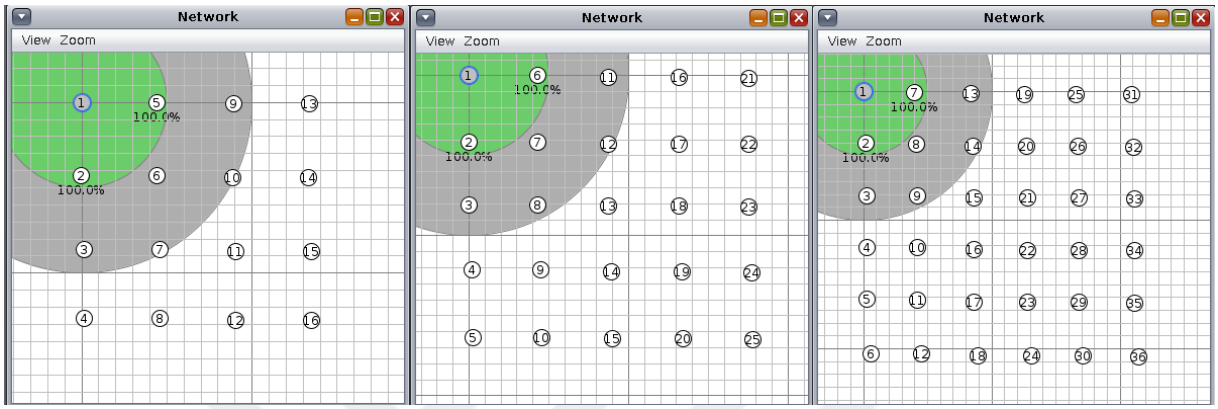


Figure 3.2.16. Topologies with 16-nodes, 25-nodes and 36-nodes for Analyses of IfNoT Mechanism

The IfNoT mechanism operates under the assumption that attacker nodes generate Interest packets with random names. In this scenario, the attacker lacks knowledge about the network and its naming conventions. As a result, the attacker consistently generates random Interest packets in an attempt to disrupt the network's functionality. In this way, the attacker node employs a traffic amplification technique to produce a substantial volume of counterfeit Interest messages. The performance analysis of the IfNoT mechanism is rooted in evaluating its efficacy within the context of this scenario. It aims to assess how well IfNoT manages and mitigates the impact of attackers employing such techniques to flood the network with counterfeit Interest messages.

The performance analysis of the IfNoT mechanism was conducted using the Cooja network simulator across various topologies. Figure 3.2.16 illustrates topologies comprising 16, 25, and 36 nodes, respectively. The simulation variables used in the analyses conducted within these topologies are detailed in Table 3.2.3. Each analysis was repeated at least five times to ensure increased accuracy, with the average values of the outcomes being taken into account.

Table 3.2.3. Simulation Variables used for Analyses of IfNoT Mechanism in Cooja Network Simulator

Variable	Value
Interest Size	55-bytes
Data Size	127-bytes
PIT Size	128
FIB Size	36
Forwarding Mechanism	RONR
RONR Prefix Length	7
IfNoT Increase Factor (ψ)	1.5
IfNoT Decrease Factor (λ)	0.995
Attacker Interval	Random between 25 – 100 ms.
Node Count	16, 25, 36
Request (Interest) Count	100 * (NodeCount-1)
PIT Timeout	3 secs.
FIB Timeout	300 secs.
Caching	Disabled
TTL	16
Neighbor Table Size	10
IfNoT Minimum Alpha (min_α)	10
Requester Interval	Random between 400 – 1100 ms.

The performance of the IfNoT mechanism was analyzed using three performance metrics. These metrics were thoughtfully chosen to offer a comprehensive perspective on the performance of the approach and its influence on network functionality. Valuable insights were gained into the effectiveness of the IfNoT approach. The first performance metric, the success rate metric, quantifies the ratio of received Data packets in response to the sent Interest packets, essentially indicating the success rate or the percentage of successful requests within the network. The second metric, the average latency metric, signifies the duration between the transmission of an Interest packet and the subsequent reception of the corresponding Data packet. This measurement exclusively considers successful requests, disregarding failed ones. The values are computed as the average across all successful requests. The third metric, the total interest traffic metric, accounts for the traffic generated by the Interest packets traversing the network. It encompasses every packet, whether self-generated or forwarded, that is sent by

each node within the network. This total also includes counterfeit Interest packets generated by the attacking node.

There are variables in the IfNoT mechanism that may have a direct impact on performance. As highlighted earlier, IfNoT relies on PIT records and their timeouts, which makes the PIT timeout variable crucial. Alongside the PIT timeout value, the IfNoT λ variable, which is utilized for reducing α, and the min_α variable, which prevents α from dropping to zero, directly influence the IfNoT mechanism's performance. In the performance analyses conducted within this study, beyond the analyses observed with default values, a detailed investigation delved into the impacts of varying the PIT timeout value, IfNoT λ value, and min_α value on the IfNoT mechanism. In the "Results and Discussions" section, the detailed performance analyses of the IfNoT mechanism are presented under the sub-section titled "IfNoT: An Interesting Flooding Mitigation Mechanism."

3.2.3. TM-RONR (TTL Modified RONR) - A TTL Modification Based Forwarding Mechanism

In the RONR forwarding mechanism, records in the FIB table are stored based on the first received Data packet. Subsequent Data packets with the same prefix only impact the update time in the FIB table without altering routing information. However, even if a Data packet had arrived earlier, it might not have followed the shortest path at that time. There could be a shorter route, especially in a mobile environment where routing information can change.

In such cases, the existing TTL (Time To Live) information in the NDN header can be utilized. By default, the TTL value for both Interest and Data packets is set to 16 when they are initially created. The TTL value of these packets is decremented by one at each intermediate node they pass through. Hence, a packet with a higher TTL value can be assumed to have come via a shorter path. Adding a TTL column to the FIB table allows the path lengths to be stored in these records. Consequently, when computing routing information, the path length will also be taken into account.

It’s expected that the proposed TM-RONR forwarding mechanism will result in obtaining shorter paths. When a node receives a Data packet, it checks the FIB table. If there’s no FIB entry, it creates one by adding the TTL information. If there’s an existing FIB entry, it compares the incoming Data’s TTL with the recorded TTL in the entry. It updates the FIB entry with the path information associated with the higher TTL value, thus maintaining information about the shorter path. TM-RONR approach is visualized in Figure 3.2.17.

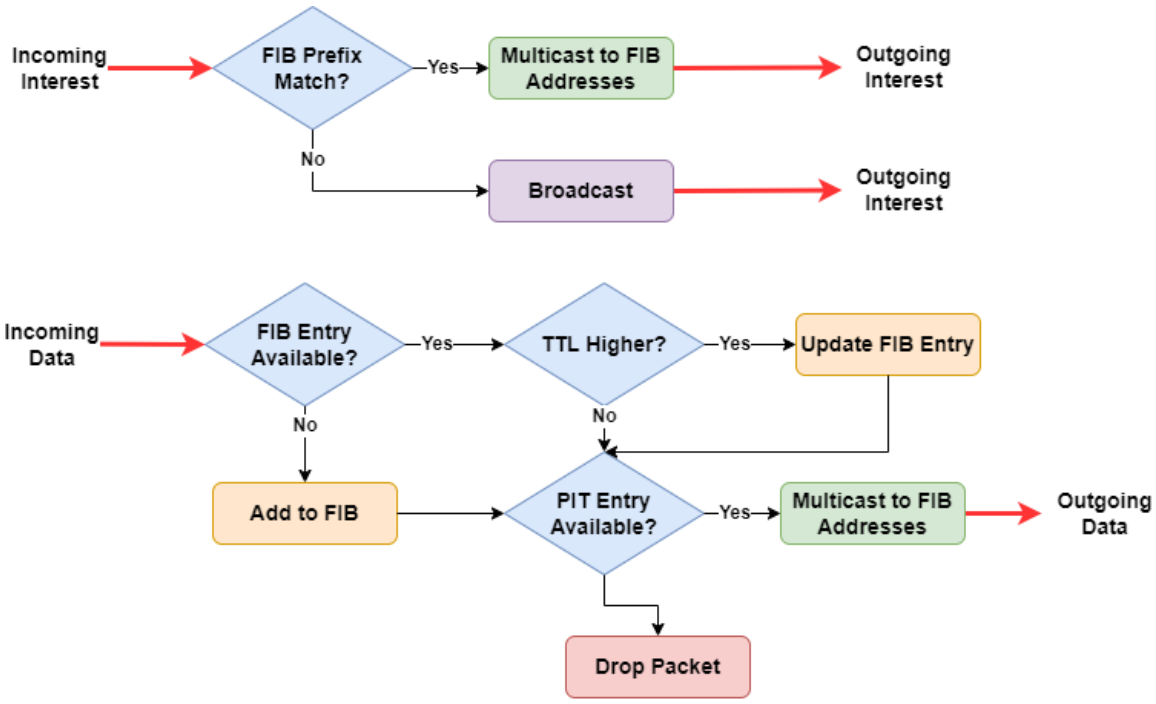


Figure 3.2.17. TM-RONR Forwarding Mechanism

TM-RONR was tested to get performance analysis within this study. Tests were conducted on two different topologies, 6x6 (36 nodes) and 7x7 (49 nodes), using the Cooja simulator environment. Figure 3.2.18 illustrates grid-type topologies. The variables and their values used in the Cooja simulation environment are listed in Table 3.2.4. Each simulation was repeated at least five times for accuracy, and the averages of these repetitions were taken. Performance analyses are performed with three metrics in mind: Success Rate, Latency, and Total Network Traffic.

- **Success Rate:** This metric represents the ratio of successfully received packets, indicating the number of Data packets corresponding to the spread of Interest packets in the network.
- **Latency:** This metric provides the observed time between Interest and Data packets. It measures the time between sending the Interest packet and receiving the corresponding Data packet.
- **Total Network Traffic:** This metric shows the total number of circulating Interest and Data packets in the network. The total size of Interest and Data packets represents the network traffic.

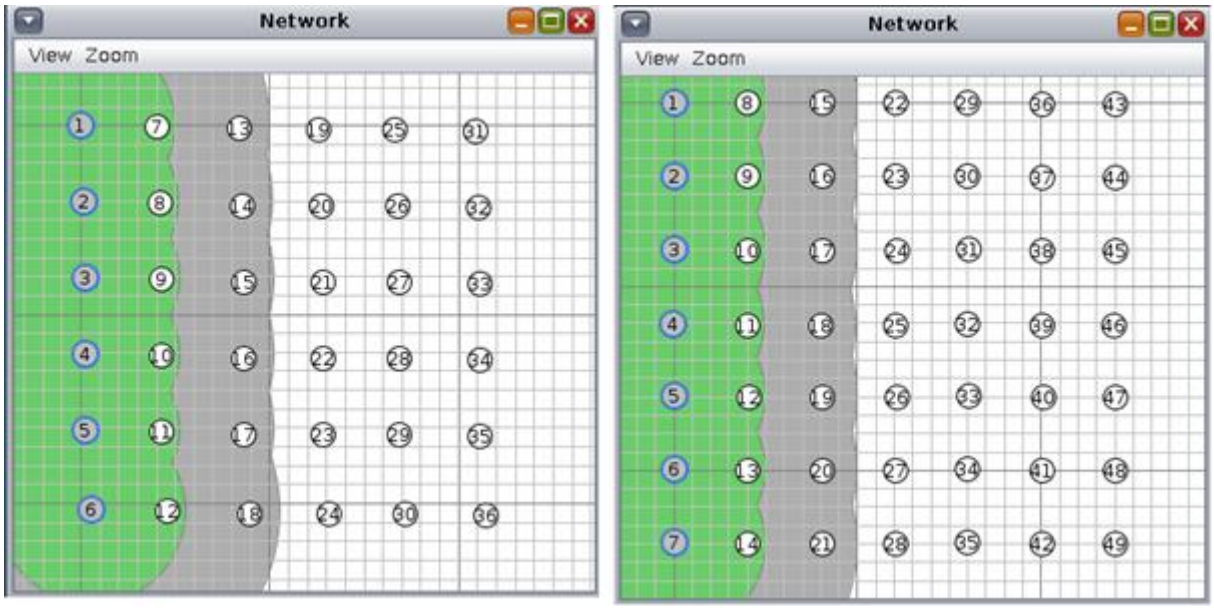


Figure 3.2.18. Grid Topology with 36-nodes and 49-nodes to Evaluate TM-RONR on Cooja Network Simulator

The comprehensive performance analysis of this study, TM-RONR, is thoroughly examined within its dedicated subsection under the Results and Discussion section.

Table 3.2.4. Simulation Variables used for RM-RONR Evaluations in Cooja Network Simulator

Variables	Values
Interest Interval	200 ms
PIT Size	30
FIB Size	40, 50
PIT Timeout	3 secs.
FIB Timeout	300 secs.
Interest Packet Size	30-bytes (excluding MAC header)
Data Packet Size	102-bytes (excluding MAC header)
Name Length	24-characters
RONR Prefix Length	7
Node Count	36, 49
Total Interest Packet Count	100 * 35 * 6, 100 * 48 * 7
Caching	Off

3.2.4. A Content Caching Analysis Study

The initial version of the μNDN protocol stack didn't include a caching mechanism. A basic caching mechanism was developed to explore its impact. With this system, each node stores a received Data packet in its Content Store, regardless of whether it has a PIT record or not. Consequently, when another node requests that Data packet, there's no need to search the entire network for the original source. An example is given in Figure 3.2.19.

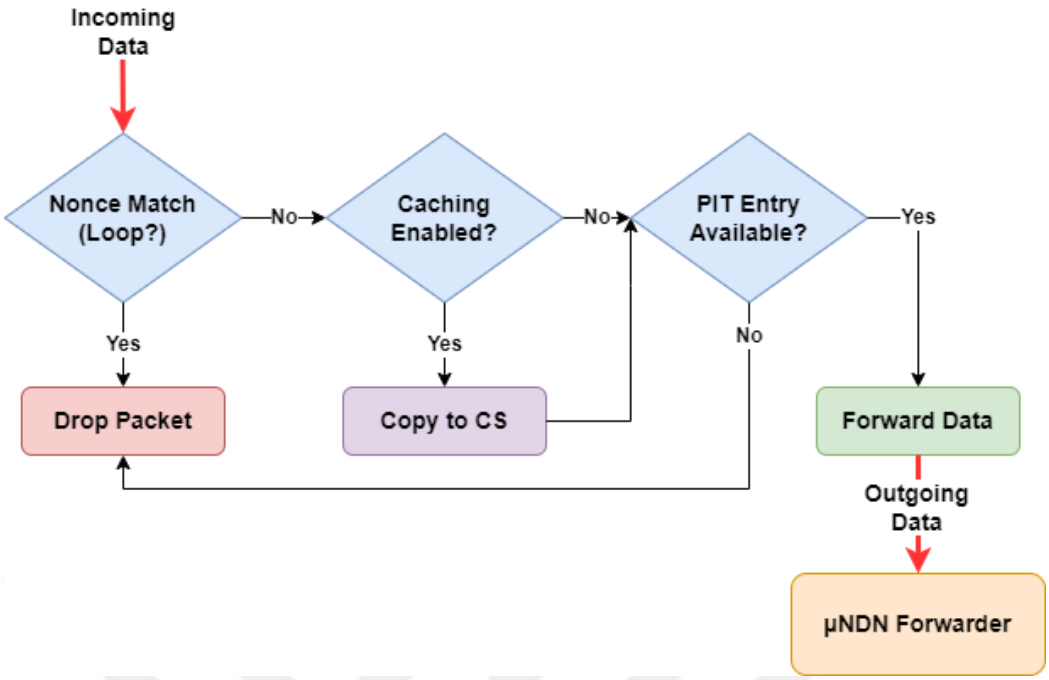


Figure 3.2.19. µNDN Caching Mechanism

A flag in the Content Store marks whether the data is cached or original, mirrored in the headers of Interest and Data packets. This flag enables nodes to choose whether to receive cached data or bypass it. This binary representation of cache status facilitates network-wide caching analysis, supporting the development of cache-aware algorithms.

Given the limited space in the Content Store database of 802.15.4 IoT devices, it's not feasible to retain every received data in the cache. Upon receiving new data, the Content Store checks for available space. If it's full, the oldest cached record gets replaced with the new data, ensuring more current Data packets remain accessible in the Content Store.

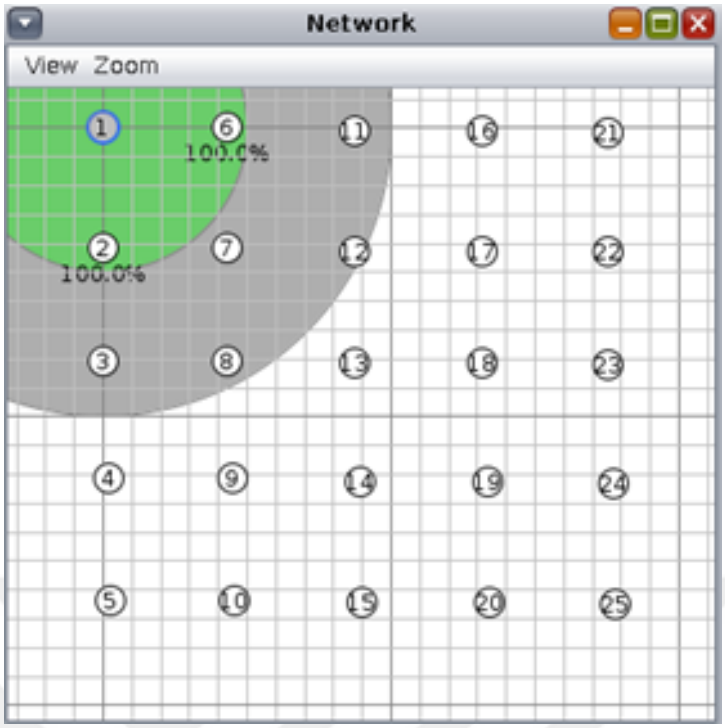


Figure 3.2.20. Grid Topology with 25-nodes for Caching Analysis on Cooja Network Simulator

The impact of the caching mechanism was evaluated through simulations conducted on grid topologies featuring 16 (4x4), 25 (5x5), and 36 (6x6) nodes. In each topology, nodes range from having a minimum of 2 to a maximum of 4 neighbors. An example of a topology with 25 nodes is given in Figure 3.2.20. The first column of nodes in the grid initiates data requests using interest packets, equating to 4, 5, and 6 nodes in the 16, 25, and 36-node topologies, respectively. Other simulation variables are provided in Table 3.2.5. Analyses are examined in 5 performance metrics. The success rate metric signifies the receipt of a data packet in response to the transmitted interest packet. The average latency metric quantifies the average duration between sending an interest packet and receiving the corresponding data packet. Total interest traffic reflects the volume of traffic generated by interest packets across the network. Total data traffic measures the traffic volume caused by data packets traversing the network. The cache hit ratio illustrates the percentage of successfully retrieved data that was obtained from the cache.

Table 3.2.5. Simulation Variables used for Caching Mechanism Analysis in Cooja Network Simulator

Variable	Value
Forwarding Strategy	RONR
RONR Prefix Size	7
PIT Size	30
FIB Size	30
Content Store Size	16, 25, 36
Nodes	24
Data size	72
MAC Protocol	CSMA

The performance outcomes from the conducted analyses are elaborated in a dedicated subsection within the Results and Discussion section.

3.2.5. An FIB Size Analysis Study

Considering the NDNoT architecture, which is the µNDN protocol stack in this study, PIT and FIB structures are the structures that play an important role in the efficient operation of µNDN. The size of the PIT and FIB tables can affect the performance of µNDN. However, overly large tables may be heavy for constrained IEEE 802.15.4 devices. For this reason, if the optimal PIT/FIB size can be determined, a more efficient µNDN protocol stack can be obtained.

In this study, exploring the optimal PIT/FIB table size, as identified through preliminary studies and literature reviews, it became apparent that the PIT table size correlates with the quantity of unique data present in the network. This makes it difficult to reconcile PIT size with different topologies and scenarios. Additionally, the PIT timeout value is also an important factor when choosing the PIT size. At the same time, PIT is a structure that is much more dynamic than FIB's.

From the preliminary studies and literature research conducted, it's been recognized that the FIB size correlates with the number of nodes within the network. Considering this situation and

the fact that FIB is less dynamic than PIT, it is concluded that the selection of optimal FIB size can be a more general research.

As previously discussed, the FIB size is intrinsically linked to the network's node count. The study encompassed tests that examined varied FIB sizes across topologies featuring different node quantities. The simulation variables used in these tests are presented in Table 3.2.6. Each simulation was repeated at least three times, and averages of these results were taken as results. The network topology in which the simulations were carried out is given in Figure 3.2.21.

Table 3.2.6. Simulation Variables used for FIB Table Size Analysis in Cooja Network Simulator

Variables	**Values**
Interest Packet Length	30-bytes (excluding MAC header)
Data Packet Length	102-bytes (excluding MAC header)
RONR Prefix Length	7
Node Count	8, 16, 24, 32, 40, 48
Requester Node	1
PIT Table Size	30
FIB Table Size	8, 16, 24, 32, 40, 48
PIT Timeout	3 seconds
FIB Timeout	300 seconds

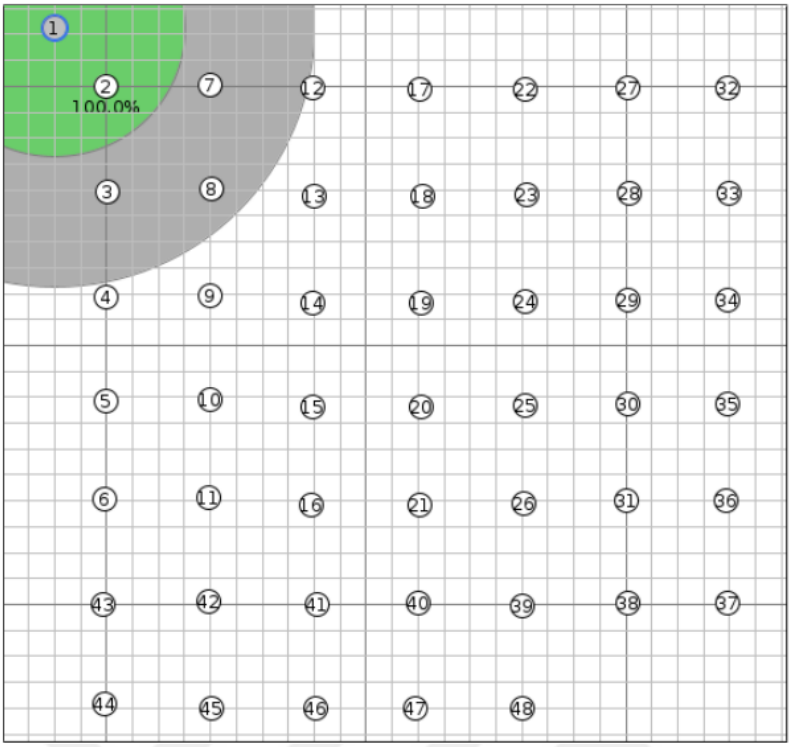


Figure 3.2.21. Grid Network Topology with 48-nodes for FIB Size Analysis on Cooja Network Simulator

Four distinct metrics were used to assess performance, outlined as follows:

- **Average FIB Occupancy:** Reflects the mean occupancy rate of the FIB table, storing forwarding information.
- **Maximum FIB Occupancy:** Illustrates the peak occupancy of the FIB table for storing forwarding details.
- **Reliability:** Indicates the success rate of received data packets relative to the sent Interest packets.

The analysis results, derived from the simulation values and variables outlined in this section, are thoroughly reviewed in the subsection titled “An Analysis of FIB Size” within the “Results and Discussion” section.

4. RESULTS AND DISCUSSIONS

Within this section, a comprehensive examination of all performance evaluations conducted within the study is presented, each analyzed in detail under their respective sub-sections. In the previous section, "Materials and Methods," detailed descriptions outline the evaluation environments, variables, and scenarios employed throughout the analyses conducted within this study.

4.1. Initial Analyses

During the initial stage, performance testing was conducted on the developed µNDN protocol stack. These tests specifically measured the performance of the VIF and RONR forwarding mechanisms after their integration into the µNDN protocol stack. Within these analyses, the evaluations were conducted utilizing the previously detailed metrics: success rate, average delay, and total network traffic. These metrics served as benchmarks for the assessments performed.

The detailed performance analysis conducted across two distinct network topologies, grid and ring, for the success rate metric, is visually represented in Figure 4.1.1. This graphical representation distinctly showcases that, across both of these network setups, the RONR forwarding mechanism consistently achieves notably higher success rates compared to the VIF forwarding mechanism. The consistently higher success rates achieved by the RONR forwarding mechanism in comparison to the VIF forwarding mechanism can be attributed to the intrinsic broadcast feature of the VIF forwarding mechanism, causing a more frequent incidence of collisions and subsequent packet loss within the network.

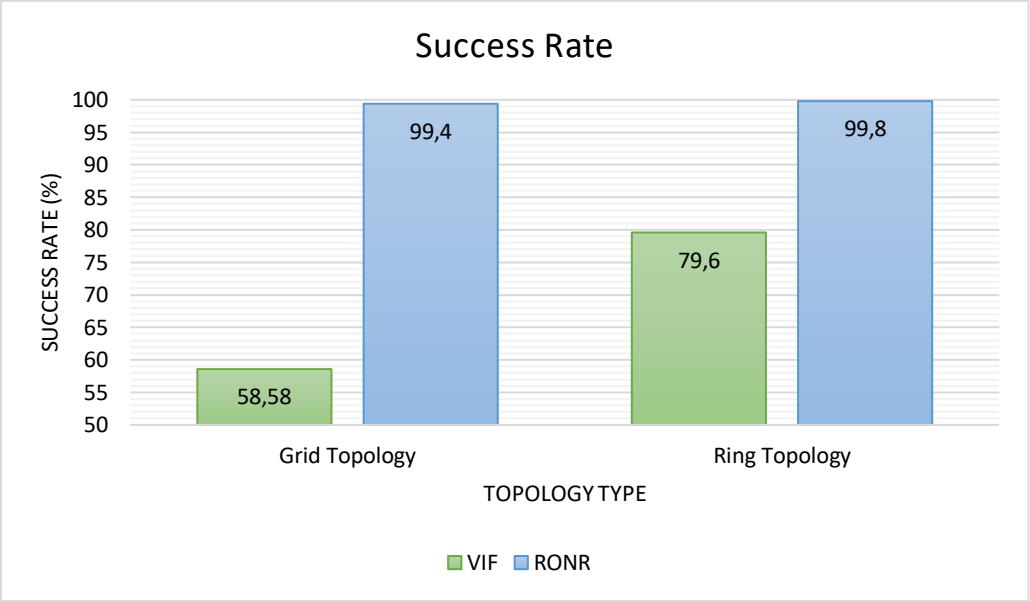


Figure 4.1.1. Success Rate Results with VIF and RONR Forwarding Mechanisms on µNDN Protocol Stack

The evaluation results for the average delay metric, with grid and ring topologies, are visually presented in Figure 4.1.2. This graphical representation distinctly indicates a noteworthy performance difference between the RONR forwarding mechanism and the VIF forwarding mechanism across both grid and ring topologies. The RONR forwarding mechanism consistently demonstrates enhanced performance when compared to the VIF forwarding mechanism in terms of average delay metrics. This trend highlights the RONR forwarding mechanism's ability to minimize delays more effectively, regardless of the network topology, suggesting its potential for offering more efficient data transmission and handling.

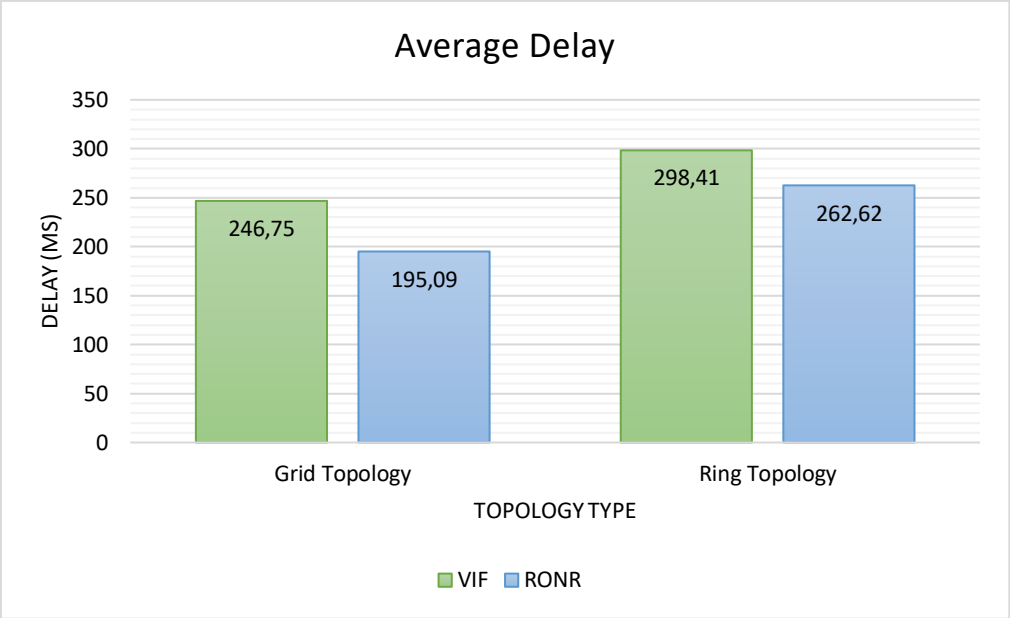


Figure 4.1.2. Average Delay Results with VIF and RONR Forwarding Mechanisms on µNDN Protocol Stack

Figure 4.1.3 represents the total network traffic metric values across both grid and ring topologies. It distinctly demonstrates that when the RONR forwarding mechanism is utilized, the overall network traffic is lower compared to the VIF forwarding mechanism. This observation implies that the RONR forwarding mechanism exhibits a higher degree of energy efficiency within the network context as it generates less network traffic.

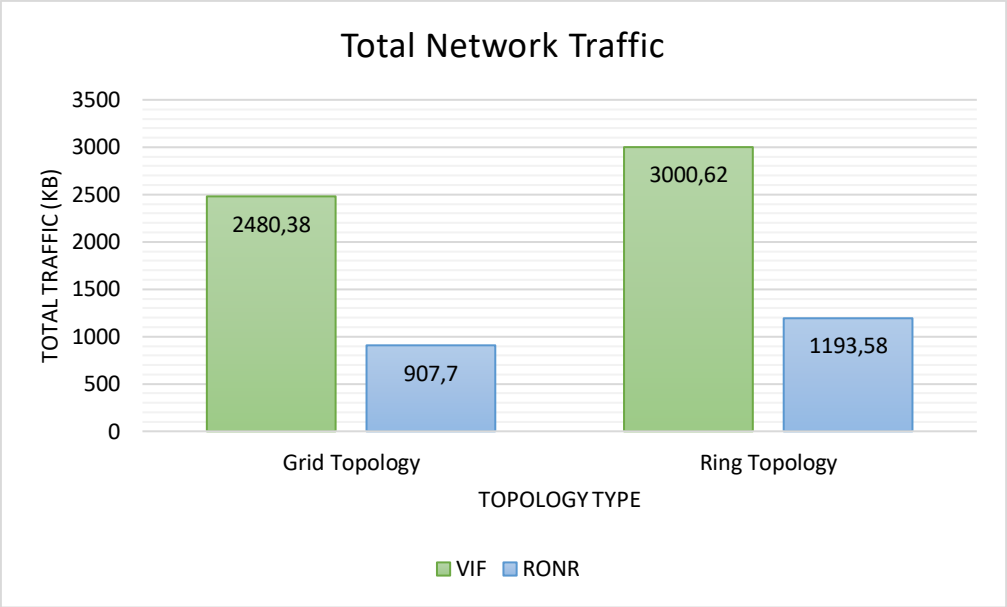


Figure 4.1.3. Total Network Traffic Results with VIF and RONR Forwarding Mechanisms on µNDN Protocol Stack

In theory, the requirement for system resources by the NDN stack is expected to be lower than that of the standard 6LoWPAN stack. An application using the 6LoWPAN stack is compared with an application using the µNDN stack to analyze this. For instance, the RPL-UDP example running on the Contiki NG operating system, which employs the 6LoWPAN stack, is considered for comparison.

Table 4.1.1 illustrates the memory consumption for both the RPL-UDP example and an example application running on the µNDN stack. The figures presented originate from the Contiki-NG operating system, compiled and constructed with the specified protocol stacks and applications. Usually, the text and data fields indicate the storage space usage (flash memory), while the bss fields signify the memory usage (RAM). These values within the table practically demonstrate that the µNDN stack consumes notably less memory and storage in comparison to the 6LoWPAN stack.

Table 4.1.1. Memory Consumption Analysis of RPL-UDP/6LoWPAN Stack Server/Client and µNDN Stack with VIF/RONR

Stack	text	data	bss
RPL-UDP Server	42435	338	5950
RPL-UDP Client	42663	338	5950
µNDN VIF	22531	216	2744
µNDN RONR	23171	216	2744

4.2. IfNoT: An Interest Flooding Mitigation Mechanism

In this section, a comprehensive examination of the IfNoT mechanism itself and its influential variables that may impact performance is conducted. Subsequent subsections focus on the performance evaluation of the IfNoT mechanism under various conditions, encompassing examinations of default variable values, the influence of the PIT timeout value, the impact of the IfNoT decrease factor, and an in-depth analysis of the IfNoT minimum alpha value on the mechanism's performance.

4.2.1. Results with Default Values

In the earlier sections, the default simulation variables while analyzing the IfNoT mechanism were outlined in Table 3.2.3. Initial tests were performed using these default values to assess the effects of adjustments in PIT Timeout, IfNoT Decrease Factor, and minimum alpha values. Simulations for each scenario were run at least five times to collect data on the outcomes. The following sections provide insights into the achieved results and their analyses.

4.2.1.1. Success Rate

The success rate graphs for the grid topologies with 16, 25, and 36 nodes are shown in Figure 4.2.1, Figure 4.2.2, and Figure 4.2.3, respectively. These graphs illustrate that toggling IfNoT between enabled and disabled modes has minimal influence on the outcomes when the attacker

remains inactive. However, when the attacker is active, a noticeable decline in the success rate is observed. This clearly demonstrates the efficacy of the IfNoT mechanism.

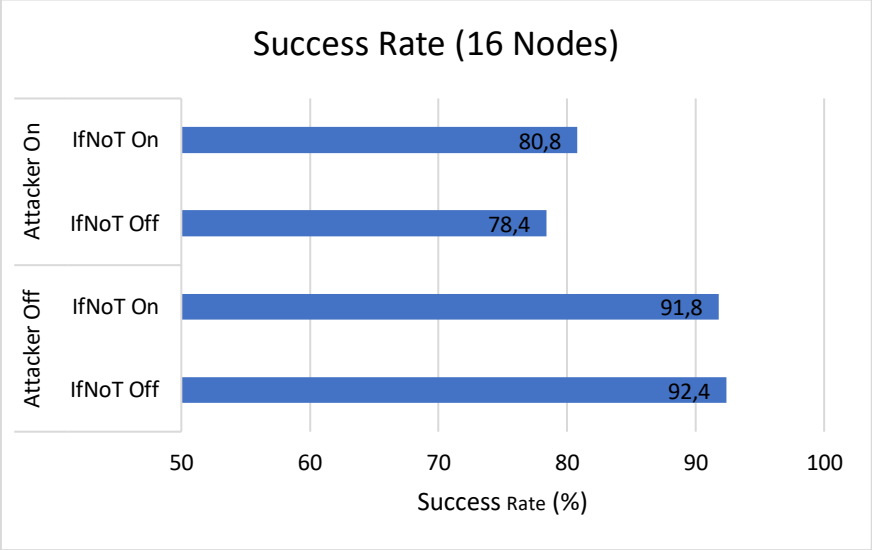


Figure 4.2.1. Success Rate Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 16-Nodes Grid Network

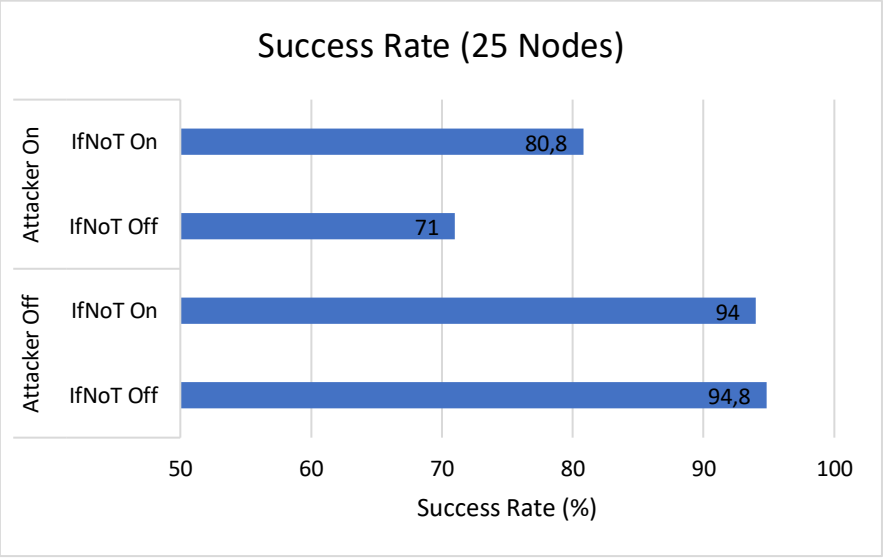


Figure 4.2.2. Success Rate Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 25-Nodes Grid Network

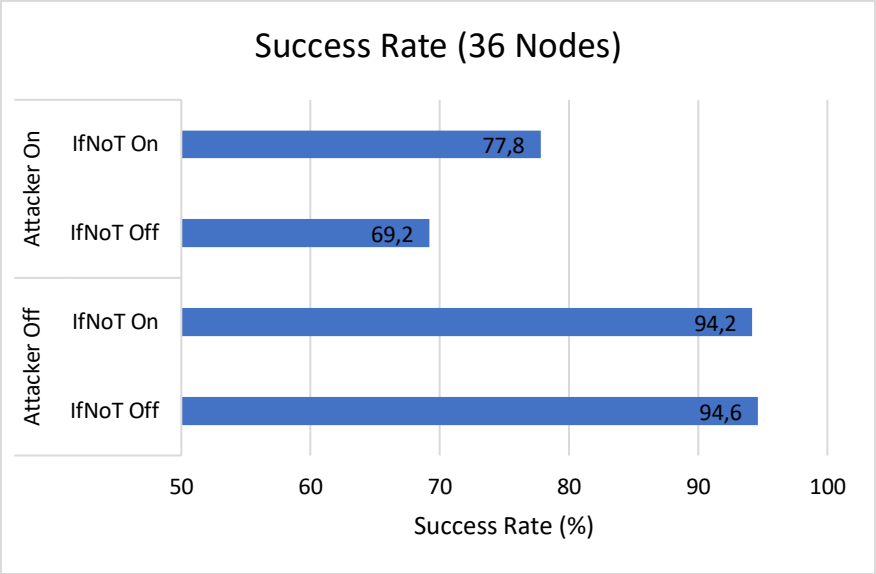


Figure 4.2.3. Success Rate Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 36-Nodes Grid Network

Notably, in densely populated networks where IfNoT is activated, this decline is less visible. This observation emphasizes how effectively IfNoT counters the negative impact of attacking nodes. For a comprehensive overview of these findings, Figure 4.2.4 represents data from all three topologies. As shown, the IfNoT mechanism enhances the success rate by up to 13% in scenarios with an active attacker.

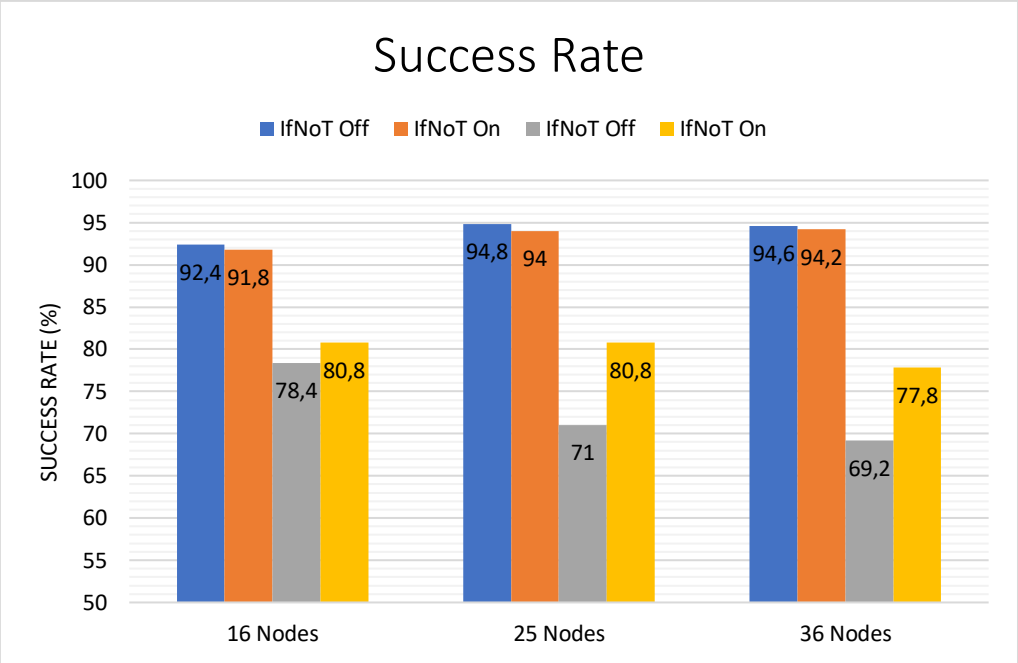


Figure 4.2.4. Success Rate Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on Different Topologies

4.2.1.2. Average Latency

Latency serves as a crucial measure in networks where the swift delivery of data is important for seamless operations. An attacker node can disrupt network operations and amplify latency by persistently generating counterfeit interest packets. Graphs in Figure 4.2.5, Figure 4.2.6, and Figure 4.2.7 clearly show how the IfNoT mechanism lowers the average latency in the presence of such attacks. Notably, IfNoT showcases the capability to decrease latency by a significant 43%. Critically, its implementation doesn't adversely impact latency when the network operates without active attacker nodes. This feature renders IfNoT suitable for deployment in networks where protection against attacks is not even an immediate concern, thereby preserving the efficiency of low-latency communication.

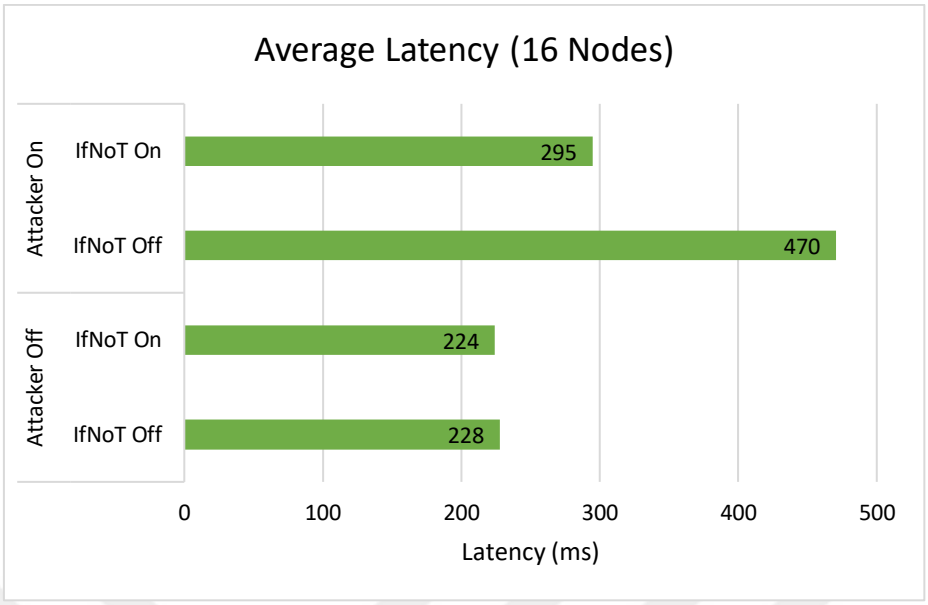


Figure 4.2.5. Average Latency Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 16-Nodes Grid Network

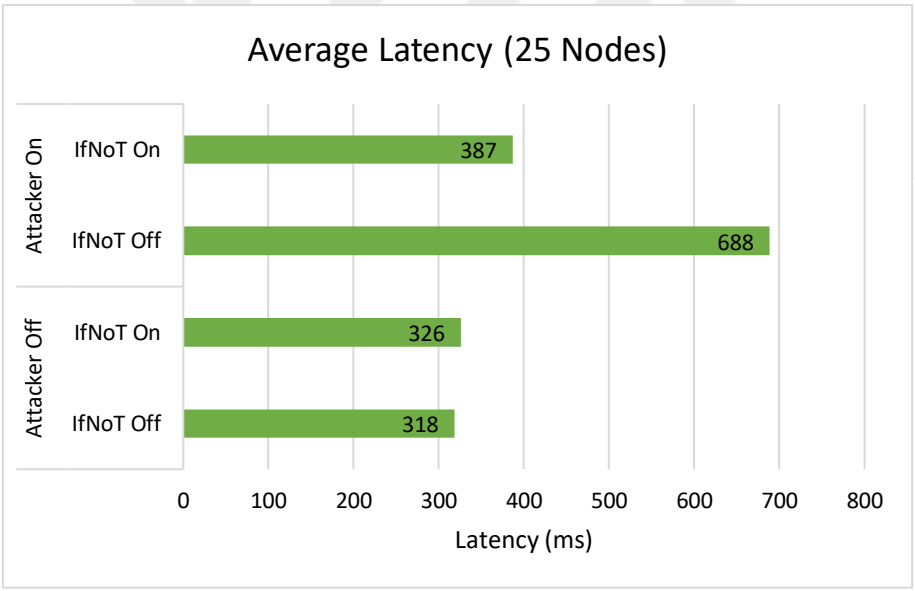


Figure 4.2.6. Average Latency Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 25-Nodes Grid Network

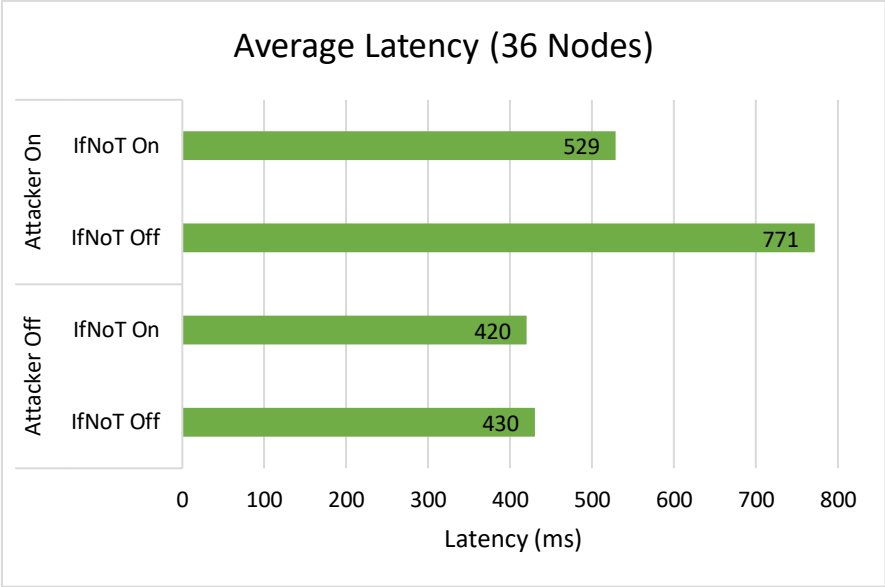


Figure 4.2.7. Average Latency Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 36-Nodes Grid Network

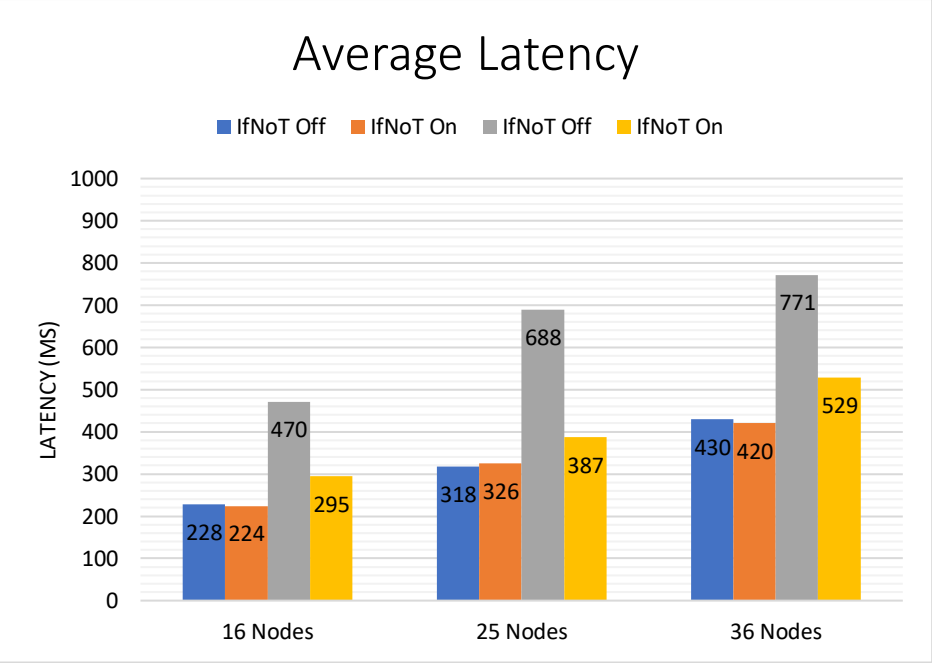


Figure 4.2.8. Average Latency Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on Different Topologies

Additionally, average latency results for all three topologies are presented in Figure 4.2.8. This graph allows the examination of average latency times across topologies with different node counts.

4.2.1.3. Total Interest Traffic

The simulation findings for the total interest traffic metric provided in graphs Figure 4.2.9, Figure 4.2.10, and Figure 4.2.11 showcase the behavior of networks containing 16, 25, and 36 nodes, respectively. Predictably, when the attacker is active, there's a noticeable increase in interest packet circulation throughout the network. However, the implementation of the IfNoT mechanism stands out for its ability to significantly decrease this heightened interest packet traffic, effectively mitigating the impact of an active attacker.

Crucially, under scenarios lacking an active attacking node, IfNoT displayed no negative influence on network performance, maintaining consistency with other performance metrics. This underscores the efficiency of IfNoT in countering the disruptive influence of attacking nodes while preserving the network's operational efficiency under regular circumstances. Furthermore, in situations where an active attacker is present, IfNoT showcases its capability to decrease total interest traffic by up to 65%.

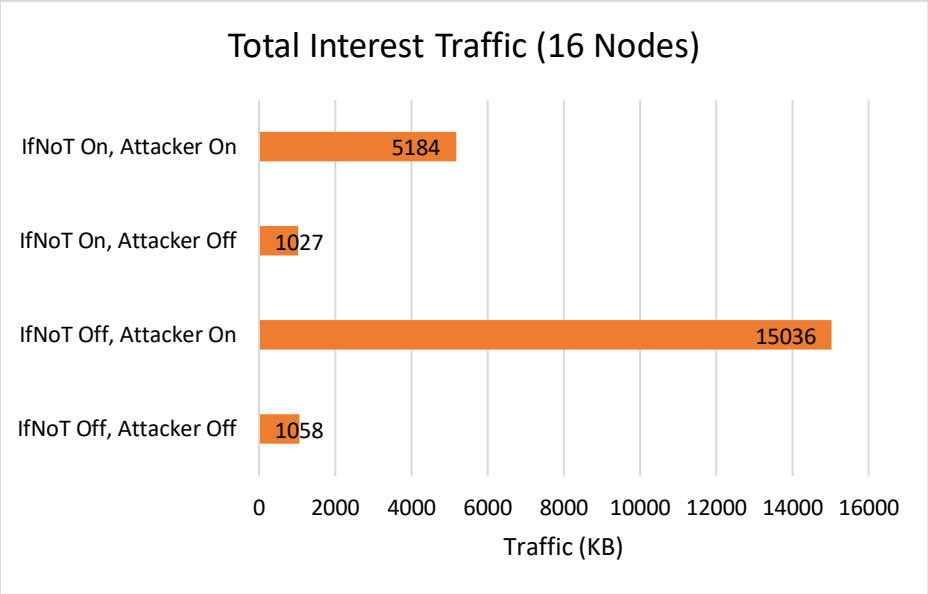


Figure 4.2.9. Total Interest Traffic Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 16-Nodes Grid Network

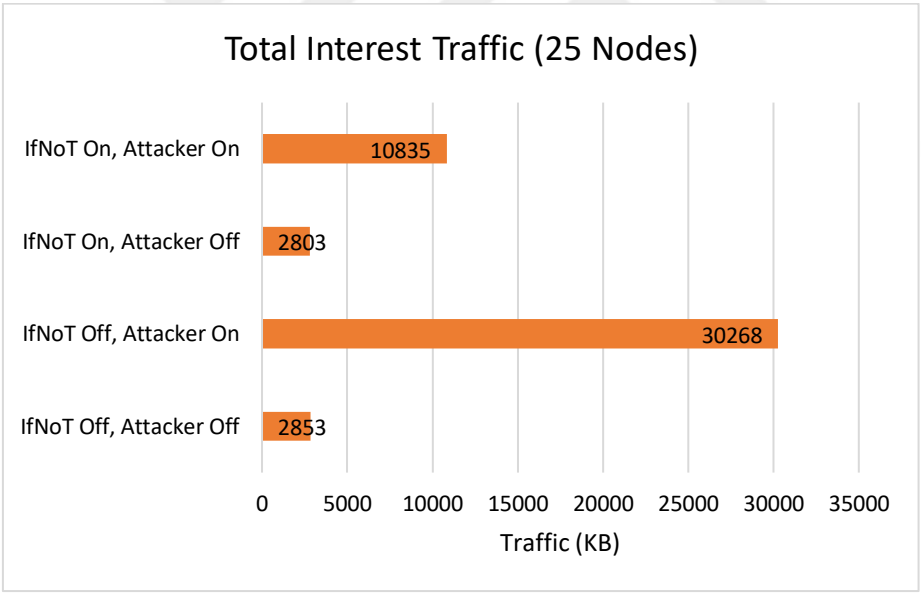


Figure 4.2.10. Total Interest Traffic Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 25-Nodes Grid Network

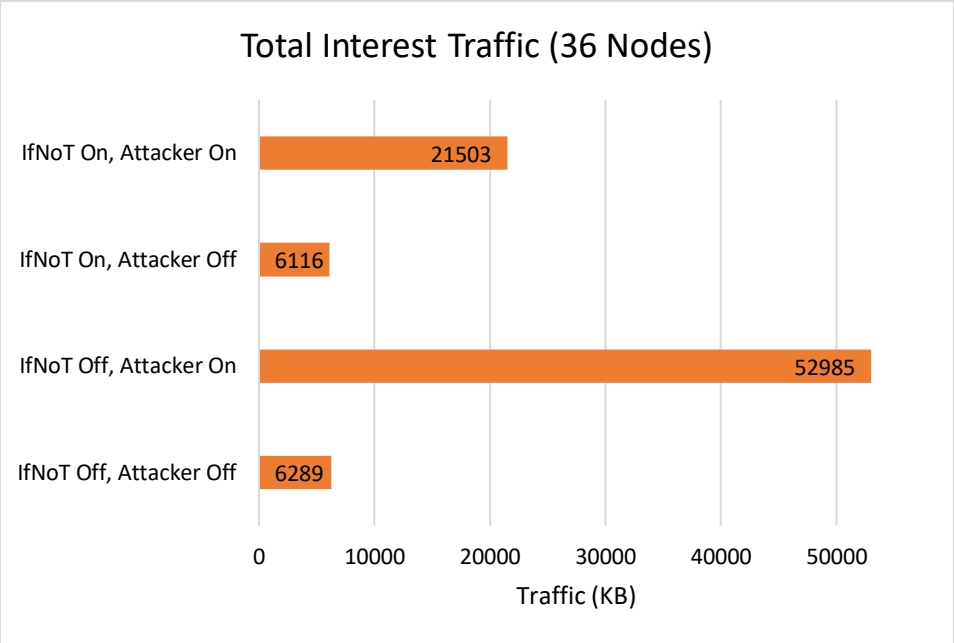


Figure 4.2.11. Total Interest Traffic Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on a 36-Nodes Grid Network

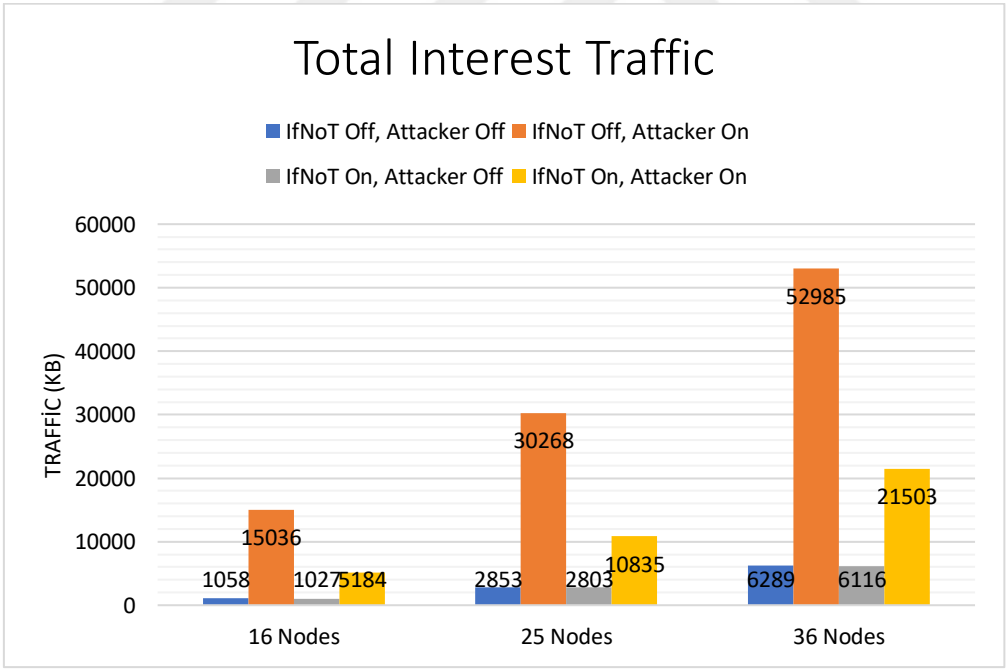


Figure 4.2.12. Total Interest Traffic Results for IfNoT Interest Flooding Mitigation Mechanism with Default Values on Different Topologies

The summary of the total interest traffic metric result for topologies with 16, 25, and 36 nodes is shown in Figure 4.2.12.

4.2.2. Results on Effect of PIT Timeout Value

As previously mentioned, the IfNoT mechanism functions based on the PIT (Pending Interest Table) timeouts. This timeout value plays a crucial role in shaping IfNoT's performance. Analyses were carried out employing different PIT timeout settings to refine results. Specifically, both the default 3-second setting and an alternative 1-second value were examined.

It's essential to highlight that pushing the PIT timeout below 1 second risks overwhelming the PIT table, resulting in operational disruptions. Hence, this range was excluded from the analysis. Conversely, extending the PIT timeout beyond 3 seconds negatively impacts performance, regardless of whether IfNoT is enabled or disabled.

Furthermore, excessively low PIT timeouts can prevent Interest packets from being fulfilled, even in the absence of an active attacker node. Consequently, the analyses concentrated on 1-second and 3-second values for practicality and to gather related insights.

4.2.2.1. Success Rate

The graphs in Figure 4.2.13 and Figure 4.2.14 illustrate the impact of different PIT timeout values (1 and 3 seconds) in scenarios with and without an IfNoT mechanism when an attacker node is active or inactive across various node configurations (16, 25, and 36 nodes). In cases without an attacker node (Figure 4.2.13), regardless of the IfNoT mechanism status, marginal differences in performance are observed between PIT timeout values of 1 and 3 seconds.

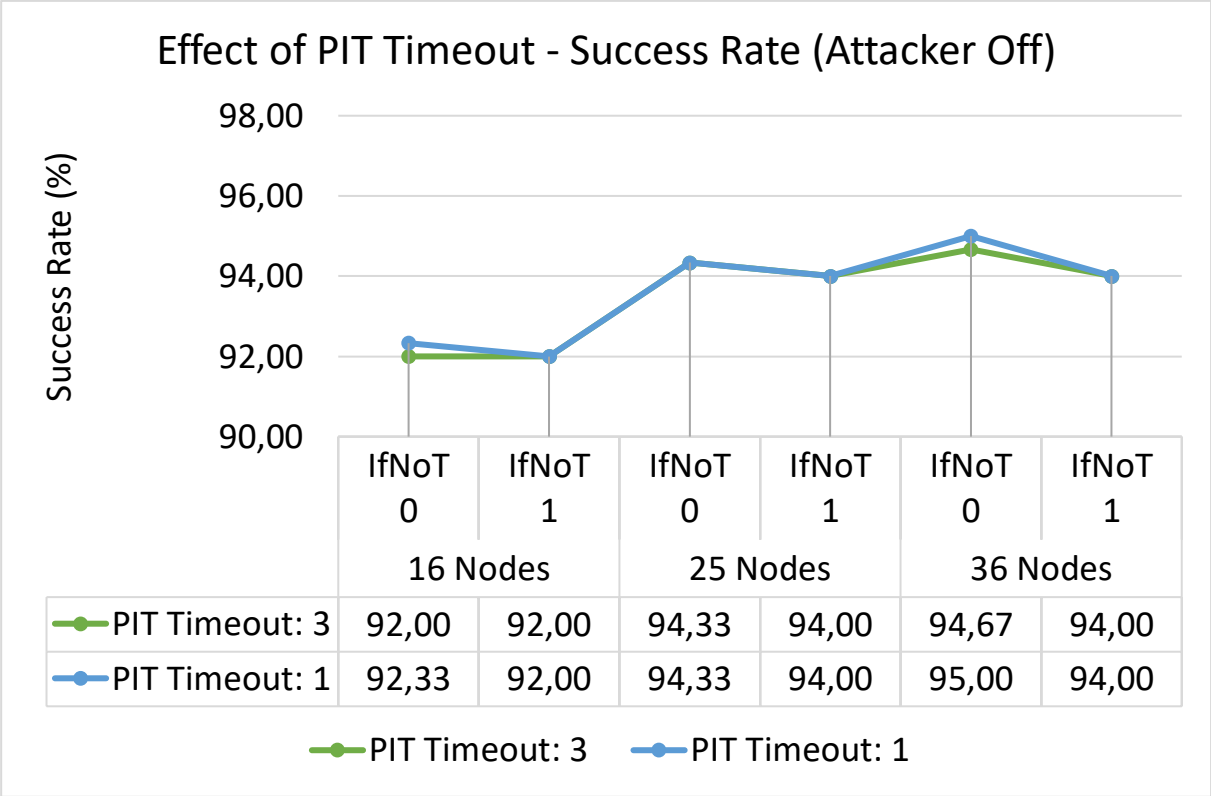


Figure 4.2.13. Success Rate Results for Different PIT Timeout Values with Enabled and Disabled IfNoT Mechanism on Different Topologies while Attacker Disabled

However, in scenarios involving an attacker node (Figure 4.2.14), a significant discrepancy in performance emerges. With an active attacker node and the IfNoT mechanism enabled, there's a substantial improvement in performance when the PIT timeout value is set to 1 second compared to 3 seconds. Specifically, with a PIT timeout of 1 second, the IfNoT mechanism demonstrates notably higher success rates across all node configurations, showcasing its ability to more efficiently detect and mitigate attacks with a shorter PIT timeout duration.

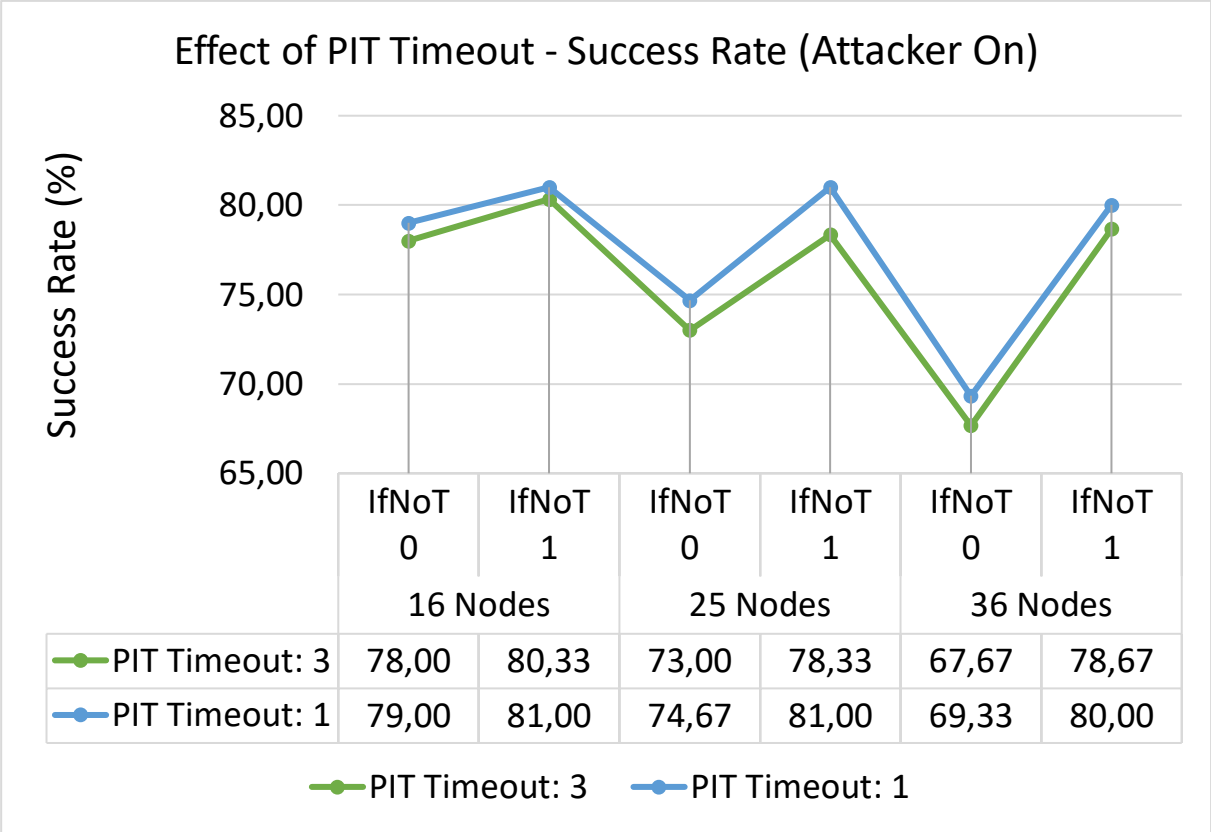


Figure 4.2.14. Success Rate Results for Different PIT Timeout Values with Enabled and Disabled IfNoT Mechanism on Different Topologies while Attacker is Enabled

4.2.2.2. Average Latency

The graphical representations in Figure 4.2.15, showcasing the influence of different PIT timeout values on the average delay metric, emphasize a clear advantage associated with a PIT timeout duration of 1 second. This graphical analysis underscores a prevalent trend favoring the 1-second PIT timeout setting, indicating notably better outcomes in average delay metrics. Despite occasional deviations, the general pattern consistently illustrates a substantial improvement when transitioning from a 3-second PIT timeout to 1 second. This adjustment consistently reflects enhanced average delay metrics across a majority of the scenarios and node configurations considered. This suggests that shorter PIT timeout values significantly contribute to minimizing average delays in network operations, serving as a valuable optimization strategy for overall network performance.

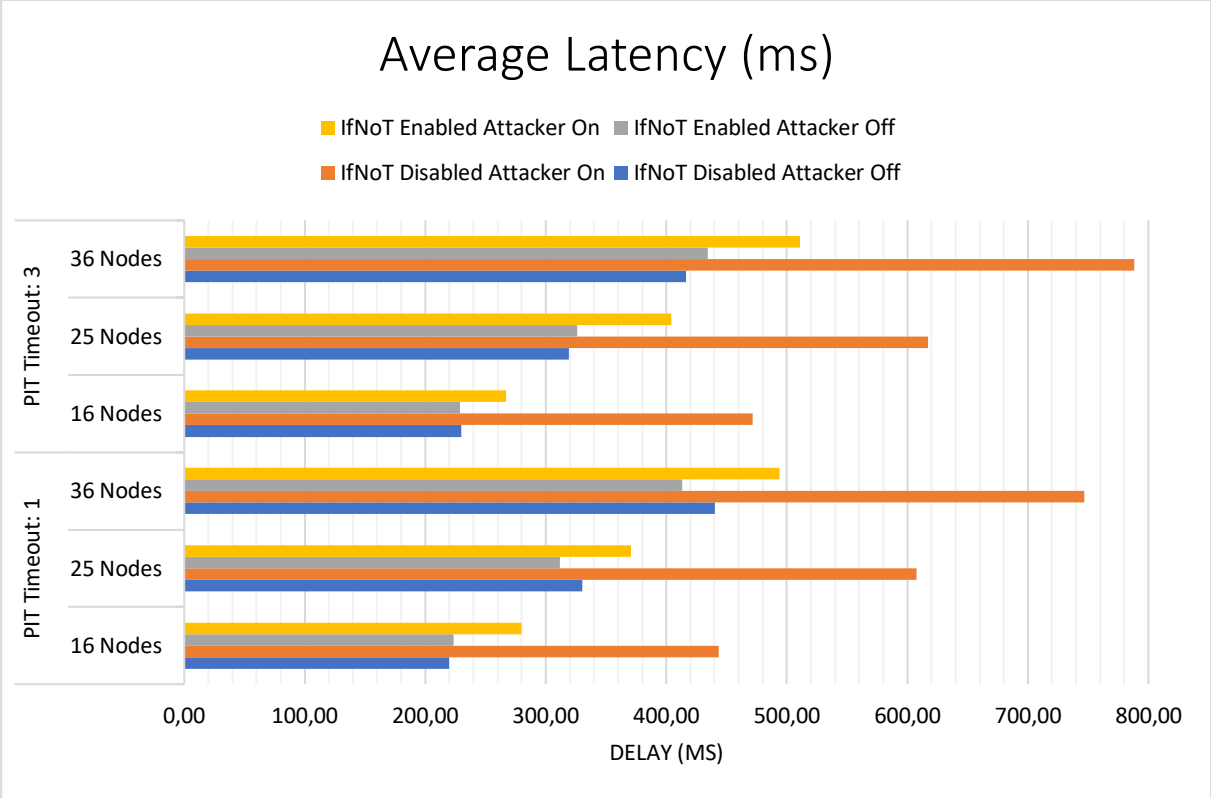


Figure 4.2.15. Average Latency Results for Different PIT Timeout Values with Enabled and Disabled IfNoT Mechanism with Enabled and Disabled Attacker on Different Topologies

4.2.2.3. Total Interest Traffic

In regard to the total interest traffic metric, the graphical representation given in Figure 4.2.16 unveils an intriguing observation: configuring the PIT timeout value to either 3 or 1 second yields remarkably comparable outcomes. The graph indicates a lack of significant disparity between these settings, emphasizing that while PIT timeout duration affects total interest traffic, the impact is not notably distinct between the 3 second and 1 second values.

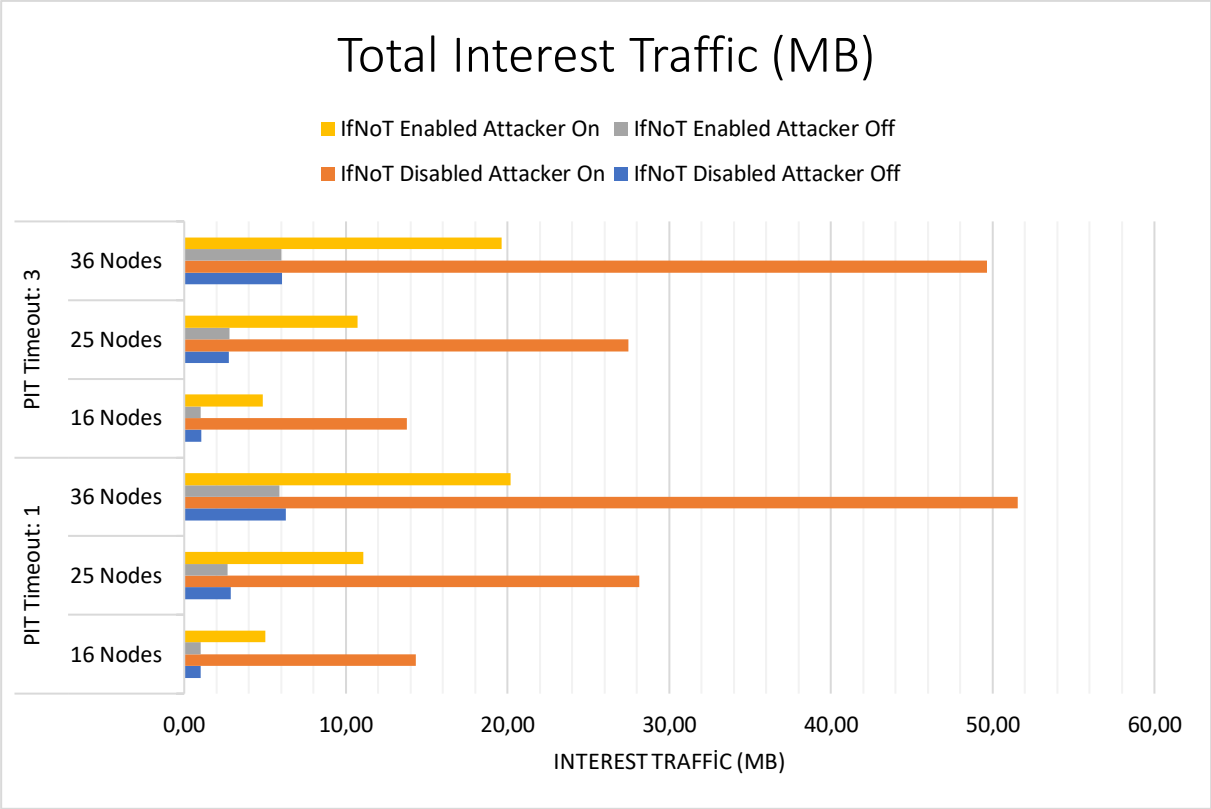


Figure 4.2.16. Total Interest Traffic Results for Different PIT Timeout Values with Enabled and Disabled IfNoT Mechanism with Enabled and Disabled Attacker on Different Topologies

Although the total interest traffic metric exhibits little distinction between the 3-second and 1-second PIT timeout settings, noteworthy enhancements are observed in both the success rate and average latency metrics when employing the 1 second PIT timeout value. Consequently, despite the lack of substantial differentiation in the total interest traffic metric, the 1 second PIT timeout emerges as the preferred default value for subsequent sections, prioritizing the overall performance improvements in success rate and average latency metrics.

4.2.3. Results on Effect of IfNoT Decrease Factor Value

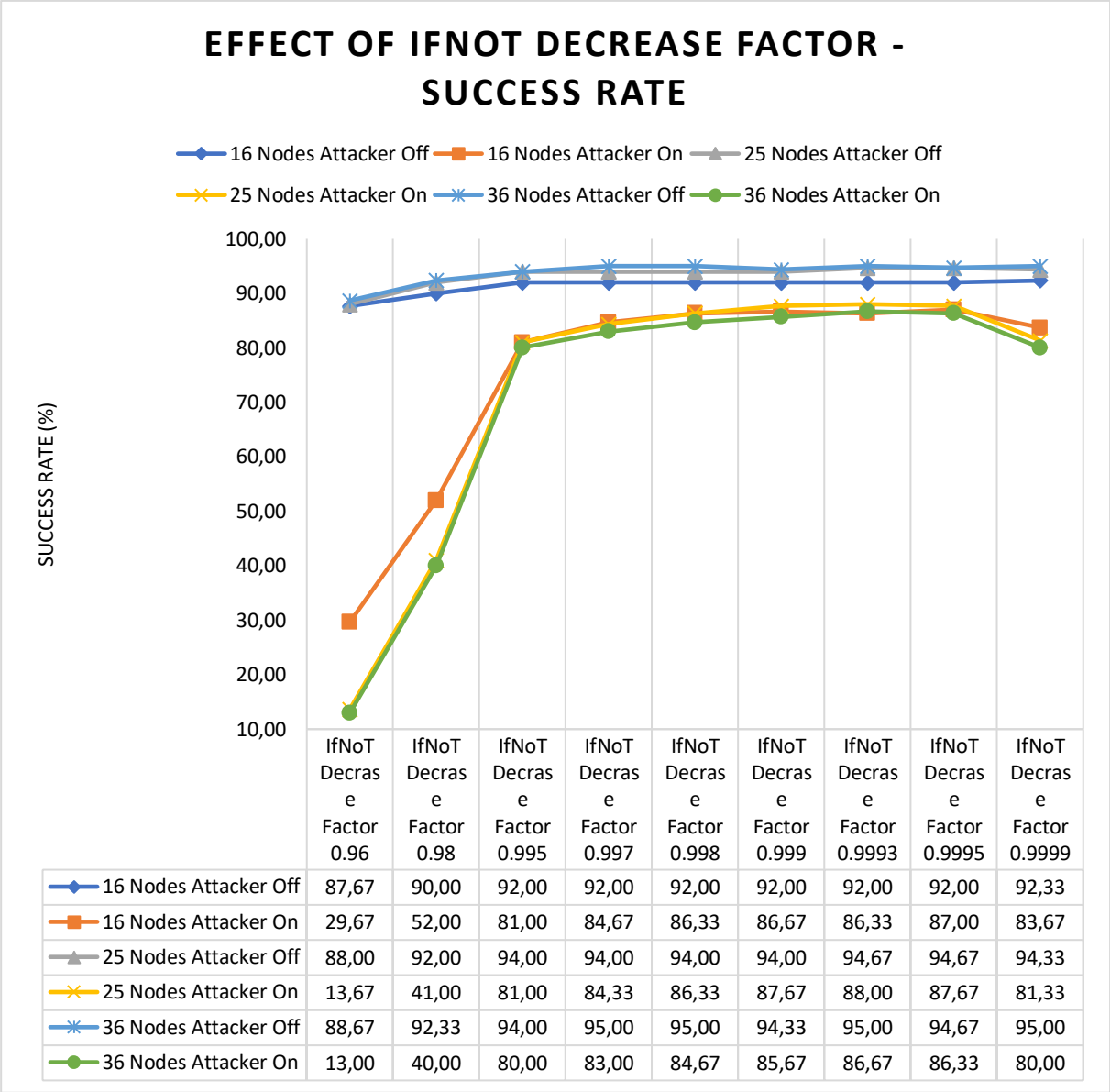
This section focuses on the impact analysis of different IfNoT decrease factor values on network performance. It encompasses various scenarios involving different node counts and attacker statuses, ranging the IfNoT decrease factors from 0.96 to 0.9999. These evaluations were

carried out across networks consisting of 16, 25, and 36 nodes, considering both attacker ‘on’ and ‘off’ conditions. The detailed assessment illustrates on how adjustments in the IfNoT decrease factor affect network behavior, particularly influencing success rate, average latency, and total interest traffic, detailed in subsequent sections.

4.2.3.1. Success Rate

As depicted in Figure 4.2.17, the success ratios were examined across different scenarios and IfNoT decrease factors (λ). Across varying node counts, the success ratios remained relatively consistent at around 90-95% when the attacker remained inactive. With higher λ an increase in success ratios was observed, particularly evident in scenarios involving 16 and 25 nodes, implying a potential positive influence of increased λ on success rates.

In scenarios where an active attacker initiated interest packet flooding, success rates experienced significant drops, ranging from approximately 13% to 86%. While higher λ demonstrated a minor increase in success rates, their impact was less pronounced compared to scenarios without an active attacker.



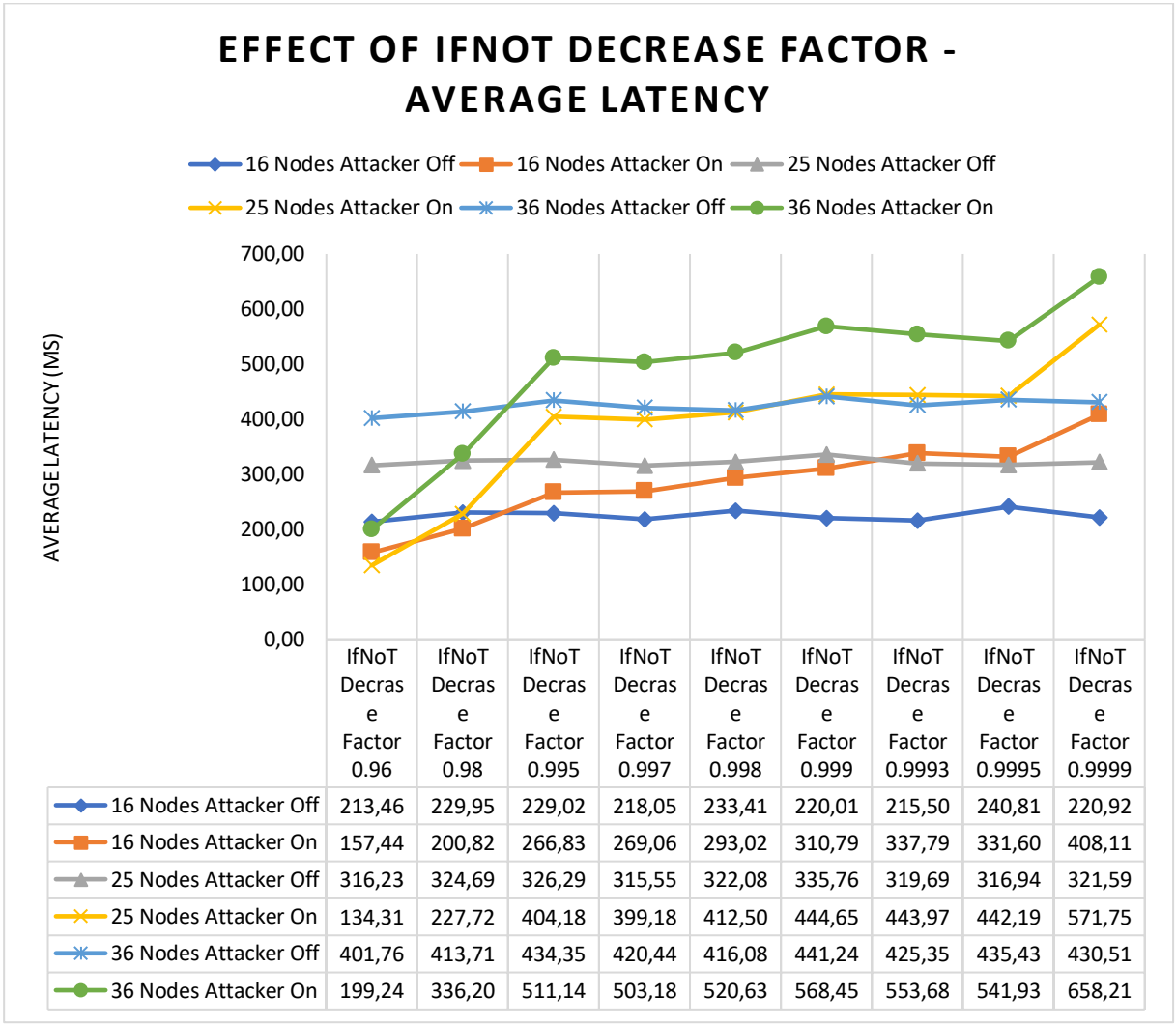
	IfNoT Decrase Factor 0.96	IfNoT Decrase Factor 0.98	IfNoT Decrase Factor 0.995	IfNoT Decrase Factor 0.997	IfNoT Decrase Factor 0.998	IfNoT Decrase Factor 0.999	IfNoT Decrase Factor 0.9993	IfNoT Decrase Factor 0.9995	IfNoT Decrase Factor 0.9999
16 Nodes Attacker Off	87,67	90,00	92,00	92,00	92,00	92,00	92,00	92,00	92,33
16 Nodes Attacker On	29,67	52,00	81,00	84,67	86,33	86,67	86,33	87,00	83,67
25 Nodes Attacker Off	88,00	92,00	94,00	94,00	94,00	94,00	94,67	94,67	94,33
25 Nodes Attacker On	13,67	41,00	81,00	84,33	86,33	87,67	88,00	87,67	81,33
36 Nodes Attacker Off	88,67	92,33	94,00	95,00	95,00	94,33	95,00	94,67	95,00
36 Nodes Attacker On	13,00	40,00	80,00	83,00	84,67	85,67	86,67	86,33	80,00

Figure 4.2.17. Success Rate Results for Effect of IfNoT Decrease Factor with Different Node Counts and Attacker Status

These findings emphasize that although higher λ might marginally enhance success ratios, the presence of an active attacker significantly undermines success rates in NDN networks, highlighting the challenge of mitigating flooding attacks.

4.2.3.2. Average Latency

The graph in Figure 4.2.18 presents the relationship between different IfNoT decrease factors (λ) and the average latency metric across varying scenarios, considering the presence or absence of an attacker node within the 16-node, 25-node, and 36-node topologies. As depicted in the graph, each λ is associated with distinct average latency values under different attacker statuses and network sizes.



	IfNoT Decrase Factor 0.96	IfNoT Decrase Factor 0.98	IfNoT Decrase Factor 0.995	IfNoT Decrase Factor 0.997	IfNoT Decrase Factor 0.998	IfNoT Decrase Factor 0.999	IfNoT Decrase Factor 0.9993	IfNoT Decrase Factor 0.9995	IfNoT Decrase Factor 0.9999
16 Nodes Attacker Off	213,46	229,95	229,02	218,05	233,41	220,01	215,50	240,81	220,92
16 Nodes Attacker On	157,44	200,82	266,83	269,06	293,02	310,79	337,79	331,60	408,11
25 Nodes Attacker Off	316,23	324,69	326,29	315,55	322,08	335,76	319,69	316,94	321,59
25 Nodes Attacker On	134,31	227,72	404,18	399,18	412,50	444,65	443,97	442,19	571,75
36 Nodes Attacker Off	401,76	413,71	434,35	420,44	416,08	441,24	425,35	435,43	430,51
36 Nodes Attacker On	199,24	336,20	511,14	503,18	520,63	568,45	553,68	541,93	658,21

Figure 4.2.18. Average Latency Results for Effect of IfNoT Decrease Factor with Different Node Counts and Attacker Status

In scenarios without an active attacker node, lower λ, such as 0.96 and 0.98, exhibit relatively higher average latency values across all topology sizes. Conversely, higher λ like 0.9993, 0.9995, and 0.9999 showcase lower average latency values in these scenarios, indicating improved network performance in mitigating interest flooding.

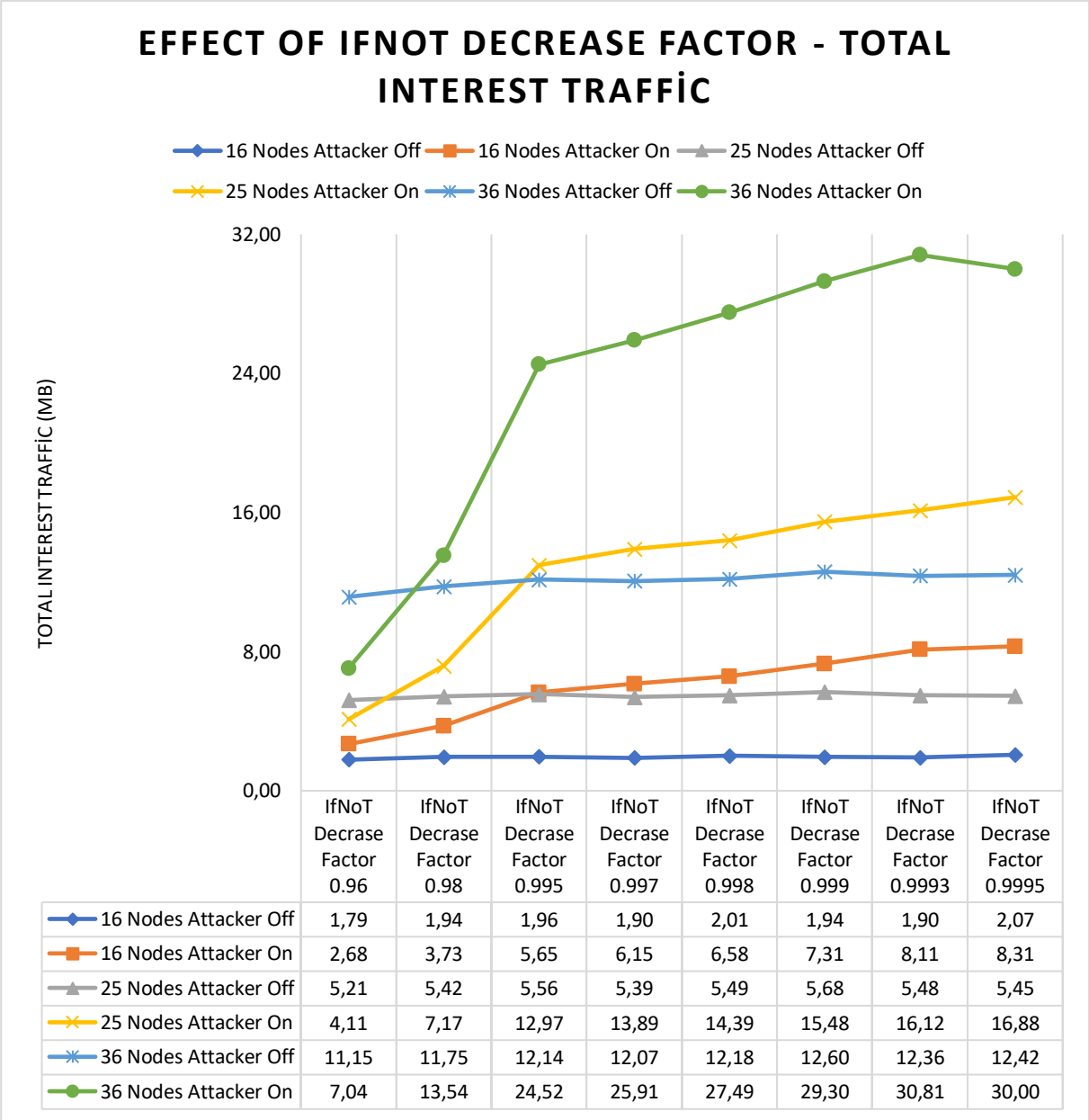
On the other hand, scenarios with an active attacker node demonstrate a contrasting trend. Higher λ result in increased average latency values, indicative of the network's struggle to manage interest flooding from the attacker. Lower λ exhibits lower latency values but might not effectively mitigate the attacker's impact.

Overall, the graph in Figure 4.2.18 visualizes how different λ's influence the average latency metric in distinct scenarios, providing valuable insights into the performance of the IfNoT mechanism under varying conditions.

4.2.3.3. Total Interest Traffic

The impact of various IfNot decrease factors (λ) on total interest traffic across different scenarios and network topologies was explored, as detailed in Figure 4.2.19. Examining the data, it's evident that different λ significantly influence the total interest traffic. For instance, at an λ of 0.9999, with 16 nodes and the attacker off, the total interest traffic shows a notable increase from 1.79 MB to 12.14 MB compared to the λ of 0.96, indicating a considerable surge in traffic with an elevated λ.

When the attacker status is 'on' across various node counts, an increase in the λ is associated with a considerable rise in total interest traffic. At 36 nodes with the attacker on, the traffic elevates from 7.04 MB at a λ of 0.96 to 41.16 MB at a λ of 0.9999, illustrating a substantial surge in interest traffic with a higher decrease factor.



	IfNoT Decrase Factor 0.96	IfNoT Decrase Factor 0.98	IfNoT Decrase Factor 0.995	IfNoT Decrase Factor 0.997	IfNoT Decrase Factor 0.998	IfNoT Decrase Factor 0.999	IfNoT Decrase Factor 0.9993	IfNoT Decrase Factor 0.9995
16 Nodes Attacker Off	1,79	1,94	1,96	1,90	2,01	1,94	1,90	2,07
16 Nodes Attacker On	2,68	3,73	5,65	6,15	6,58	7,31	8,11	8,31
25 Nodes Attacker Off	5,21	5,42	5,56	5,39	5,49	5,68	5,48	5,45
25 Nodes Attacker On	4,11	7,17	12,97	13,89	14,39	15,48	16,12	16,88
36 Nodes Attacker Off	11,15	11,75	12,14	12,07	12,18	12,60	12,36	12,42
36 Nodes Attacker On	7,04	13,54	24,52	25,91	27,49	29,30	30,81	30,00

Figure 4.2.19. Total Interest Traffic Results for Effect of IfNoT Decrease Factor with Different Node Counts and Attacker Status

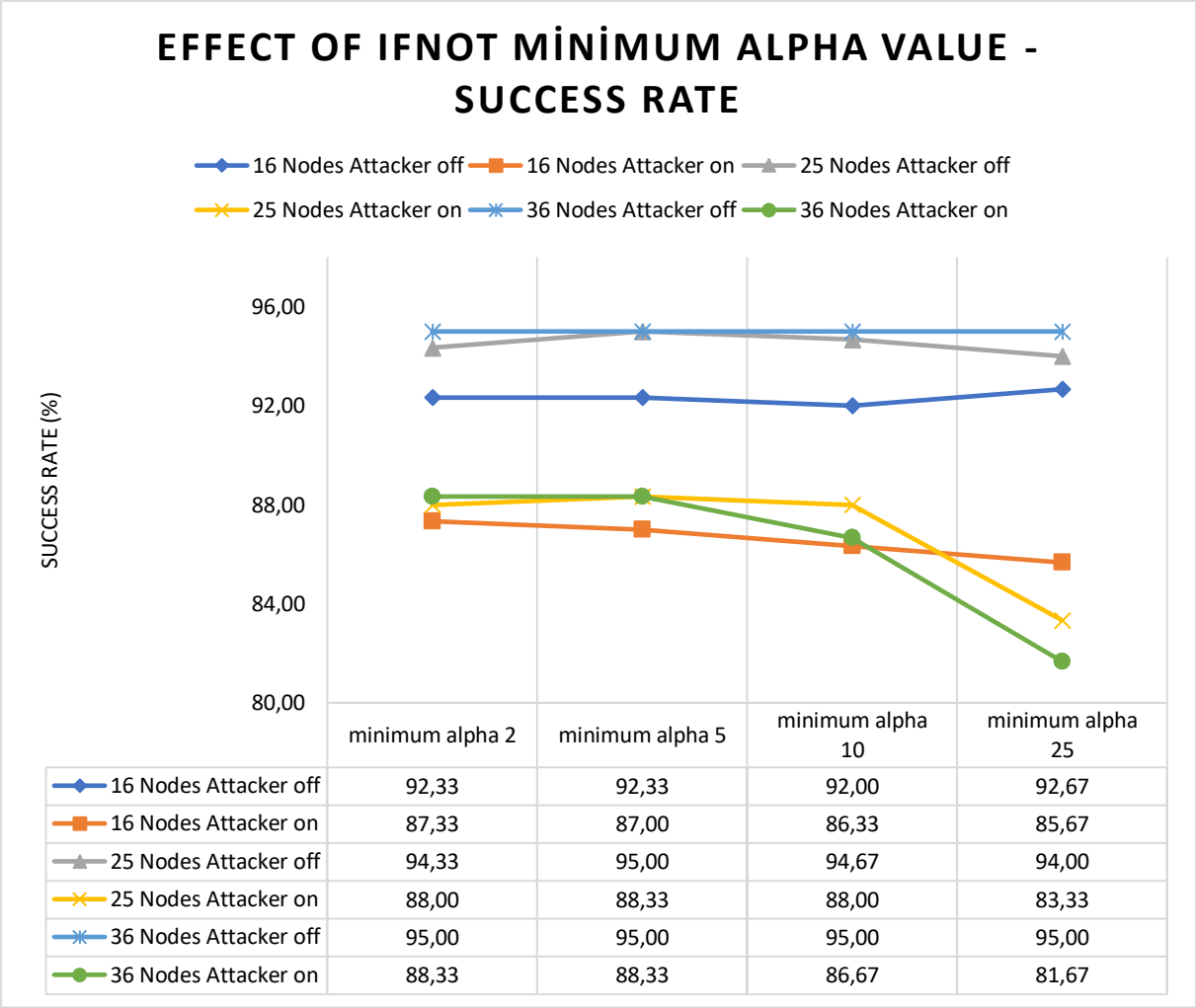
These results underscore the sensitivity of total interest traffic to changes in the λ, particularly when the attacker status is active. It's crucial to maintain a balance between the λ and the network's protective protocols to effectively manage and regulate interest traffic, as shown in the provided Figure 4.2.19.

4.2.4. Results on Effect of IfNoT Minimum Alpha Value

In exploring the effectiveness of the IfNoT mechanism as an interest flooding mitigation strategy within NDN, comprehensive evaluations were conducted across diverse scenarios. The performance of the IfNoT mechanism is notably influenced by a critical variable named the IfNoT Minimum Alpha (min_α), which was evaluated across different values: 2, 5, 10, and 25 under different performance metrics.

4.2.4.1. Success Rate

Success rate results for the effect of min_α with different node counts and attacker statuses are given in Figure 4.2.20. In examining the outcomes of these evaluations, a clear correlation emerged between the min_α variable and the IfNoT mechanism's performance. Notably, scenarios with an inactive attacker status revealed a consistent trend of higher success ratios. For instance, at a min_α of 25, the success ratio remained robustly high across all network topologies: 92.67% for 16 nodes, 94.00% for 25 nodes, and 95.00% for 36 nodes.



	minimum alpha 2	minimum alpha 5	minimum alpha 10	minimum alpha 25
16 Nodes Attacker off	92,33	92,33	92,00	92,67
16 Nodes Attacker on	87,33	87,00	86,33	85,67
25 Nodes Attacker off	94,33	95,00	94,67	94,00
25 Nodes Attacker on	88,00	88,33	88,00	83,33
36 Nodes Attacker off	95,00	95,00	95,00	95,00
36 Nodes Attacker on	88,33	88,33	86,67	81,67

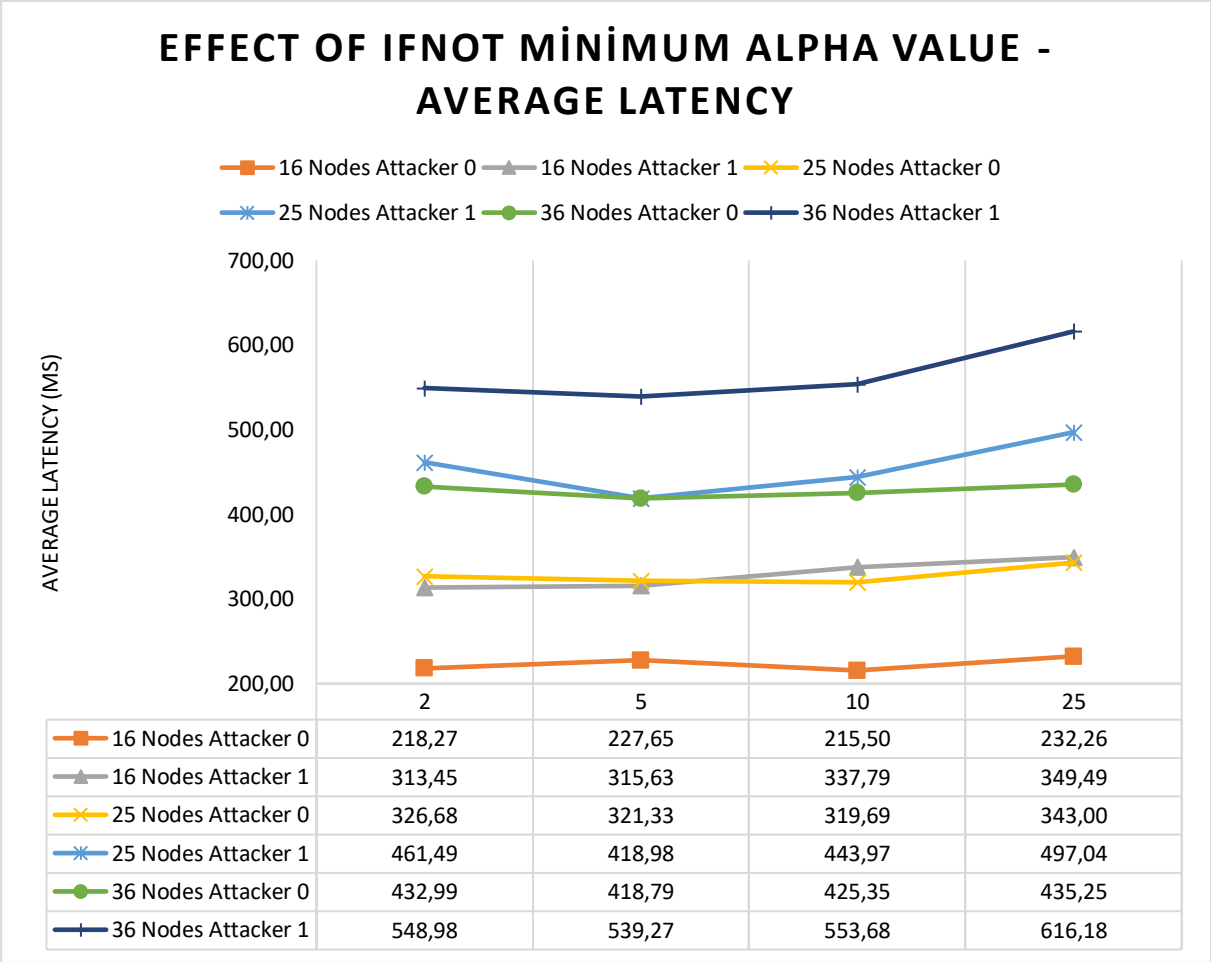
Figure 4.2.20. Success Rate Results for Effect of IfNoT Minimum Alpha with Different Node Counts and Attacker Status

Conversely, under an active attacker status, the success ratios exhibited a substantial decline across various min_α configurations. Particularly evident at the lowest min_α of 25, the success ratios decreased notably, indicating reduced efficacy in mitigating interest flooding incidents: 85.67% for 16-nodes, 83.33% for 25-nodes, and 81.67% for 36-nodes. This suggests that while higher min_α values are generally more effective, in active attack scenarios, a min_α of 10 demonstrates relatively better performance compared to other configurations, maintaining a more favorable success ratio.

These findings underscore the critical importance of optimum min_α configurations for enhanced resilience against interest flooding attacks within NDN. Figure 4.2.20 visually encapsulates the comprehensive analysis, showcasing the pivotal role played by the min_α variable in shaping the IfNoT mechanism's performance.

4.2.4.2. Average Latency

Average latency results for the effect of min_α with different node counts and attacker statuses are given in Figure 4.2.21. Upon inspecting the results of these evaluations, discernible patterns surfaced concerning the relationship between the min_α variable and average latency. In scenarios where the attacker status was inactive (off), slight changes in average latency were apparent across distinct min_α configurations. For instance, at a min_α of 25, the average latency varied among network topologies: 232.26 milliseconds for 16 nodes, 343.00 milliseconds for 25 nodes, and 435.25 milliseconds for 36 nodes.



	2	5	10	25
16 Nodes Attacker 0	218,27	227,65	215,50	232,26
16 Nodes Attacker 1	313,45	315,63	337,79	349,49
25 Nodes Attacker 0	326,68	321,33	319,69	343,00
25 Nodes Attacker 1	461,49	418,98	443,97	497,04
36 Nodes Attacker 0	432,99	418,79	425,35	435,25
36 Nodes Attacker 1	548,98	539,27	553,68	616,18

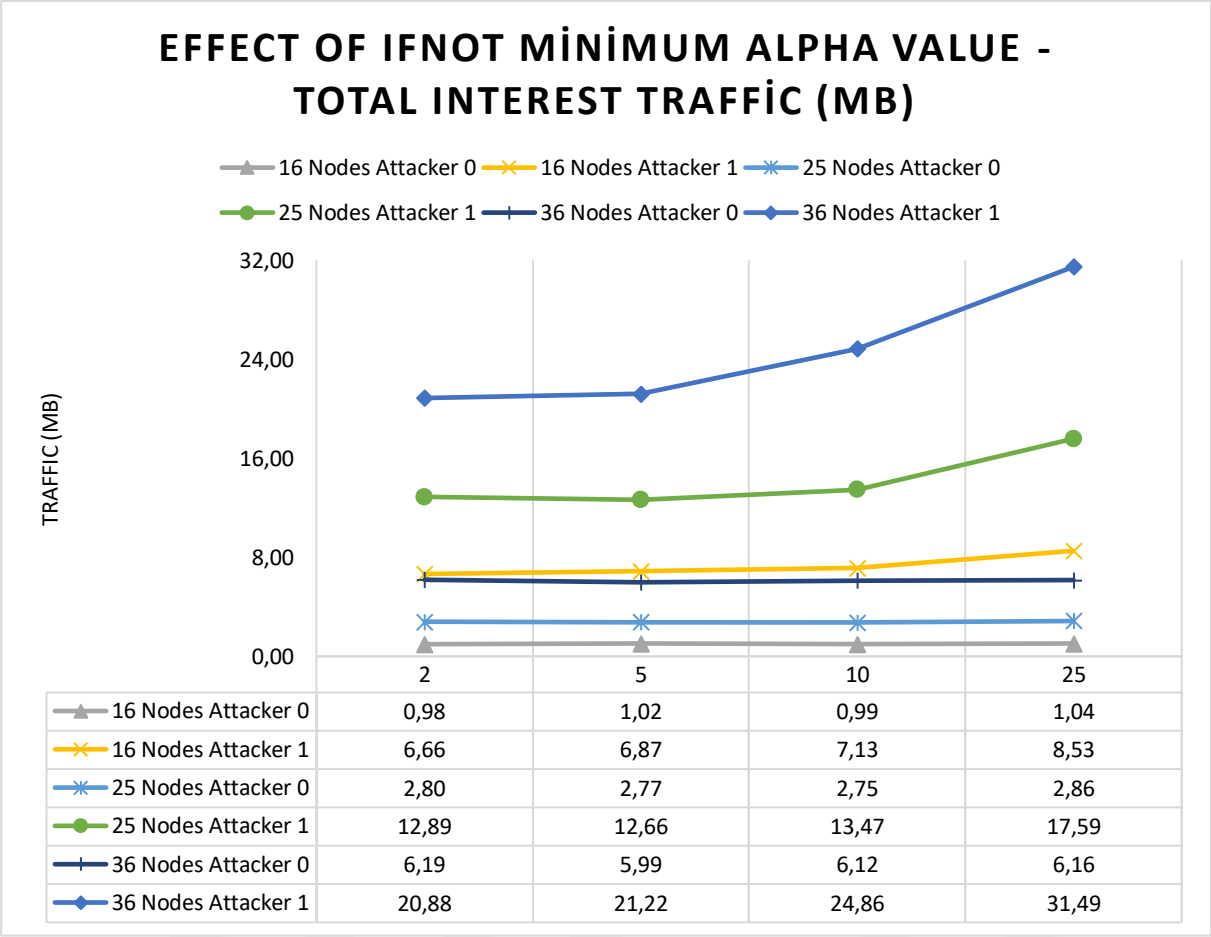
Figure 4.2.21. Average Latency Results for Effect of IfNoT Minimum Alpha with Different Node Counts and Attacker Status

Conversely, under an active attacker status (on), more changes in average latency were observed across various min_α configurations. Notably, at the lowest min_α of 25, a substantial increase in average latency was notable across all network topologies: 349.49 milliseconds for 16-nodes, 497.04 milliseconds for 25-nodes, and 616.18 milliseconds for 36-nodes. This significant rise in latency during active attack scenarios distinctly correlated with lower min_α values. Remarkably, the evaluation revealed that a min_α of 10 demonstrated relatively better performance in mitigating latency during active attack scenarios. Specifically, at this min_α value, the observed average latency showcased a more favorable outcome compared to other configurations: 337.79 milliseconds for 16 nodes, 443.97 milliseconds for 25 nodes, and 553.68 milliseconds for 36 nodes.

These findings show the impact of the min_α variable on average latency within named data networks. The graphical representation in Figure 4.2.21 visualizes the comprehensive analysis and proves the importance of the role played by the min_α variable in shaping the observed average latency in the performance of the IfNoT mechanism.

4.2.4.3. Total Interest Traffic

The graphical representation for the total interest traffic metric is given in Figure 4.2.22. A clear result is evident, showing that lower values of the min_α parameter consistently align with reduced total interest traffic across various situations. This supports the idea that setting a lower min_α level contributes to reducing overall network traffic significantly. The graph effectively demonstrates this relationship, emphasizing how lower min_α values directly lead to decreased total interest traffic, highlighting the effectiveness of employing lower min_α configurations to achieve traffic reduction goals within the network structure.



	2	5	10	25
16 Nodes Attacker 0	0,98	1,02	0,99	1,04
16 Nodes Attacker 1	6,66	6,87	7,13	8,53
25 Nodes Attacker 0	2,80	2,77	2,75	2,86
25 Nodes Attacker 1	12,89	12,66	13,47	17,59
36 Nodes Attacker 0	6,19	5,99	6,12	6,16
36 Nodes Attacker 1	20,88	21,22	24,86	31,49

Figure 4.2.22. Total Interest Traffic Results for Effect of IfNoT Minimum Alpha with Different Node Counts and Attacker Status

Furthermore, the presence of an attacker node, denoted as Attacker 1, consistently elevated total interest traffic in comparison to scenarios lacking an attacker, labeled as Attacker 0. This contrast emphasizes the considerable impact of attacks in generating additional interest traffic within the network. Consequently, it substantiates the necessity for robust mitigation strategies, wherein the configuration of min_α plays a crucial role in managing and potentially curbing the augmented traffic influx resulting from malicious attacks.

Lower min_α values (specifically 2 or 5) should be preferred when aiming to minimize network traffic, as these configurations typically result in reduced total interest traffic, especially in scenarios where an attacker node is present within the network.

4.3. TM-RONR (TTL Modified RONR)

In a comprehensive evaluation of network performance within NDN architectures, an exploration of two forwarding mechanisms, Default RONR and TM-RONR, was conducted across distinct grid topologies comprising 36 nodes and 49 nodes. The assessment focused on three crucial metrics: success rate, average latency, and total network traffic to evaluate the comparative efficiencies of these forwarding mechanisms. Each metric's evaluation served as a pivotal aspect in understanding the mechanisms' performance under varying network sizes. The subsequent analysis shows the results observed in these metrics, exposing the mechanism's operational differences in these simulated network environments.

4.3.1. Success Rate

The results for the success rate metric are given in Figure 4.3.1. In the scenario using the Default RONR forwarding mechanism, for both the 36-node and 49-node grid topologies, the success rate was 82.8% and 73.2%, respectively. On the other hand, with the TM-RONR forwarding mechanism, the success rates were recorded at 83% for the 36-node grid and remained unchanged at 73.2% for the 49-node grid.

Comparing the performance between the Default RONR and TM-RONR mechanisms, it appears that the TM-RONR slightly outperformed the Default RONR in the 36-node grid topology by a marginal 0.2%. However, in the 49-node grid topology, both forwarding mechanisms exhibited the same success rate of 73.2%.

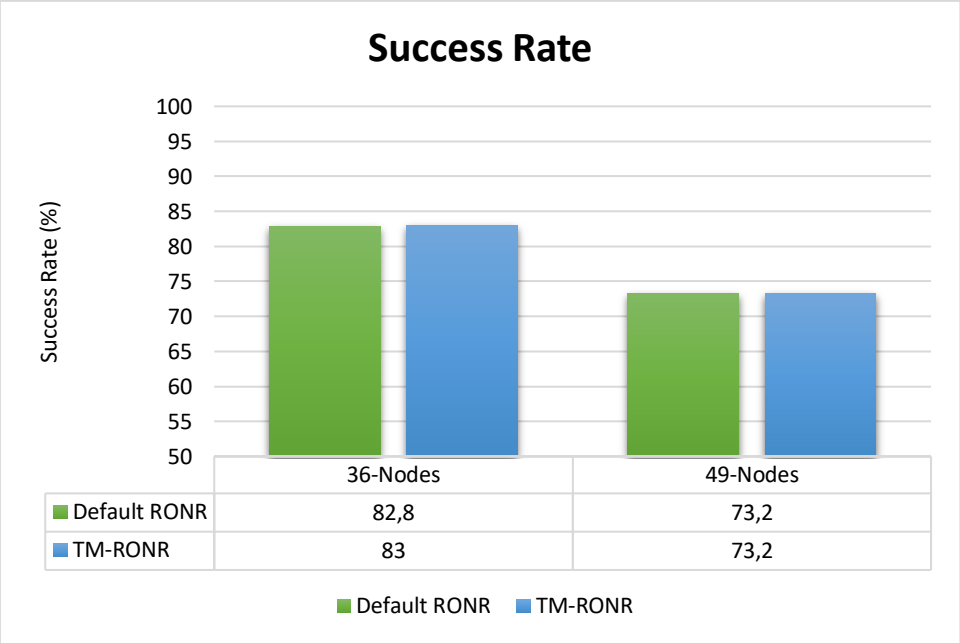


Figure 4.3.1. Success Rate Results with Default RONR Forwarding Mechanism and TM-RONR Forwarding Mechanism under Topologies with 36-nodes and 49-nodes

This suggests that while the TM-RONR shows a slight improvement in success rate over the Default RONR in the smaller 36 node grid, their performance remains identical in the larger 49 node grid. Further analysis of additional metrics may provide a more comprehensive understanding of their comparative performance in different network scenarios.

4.3.2. Average Latency

The average latency metric result graph for Default RONR and TM-RONR is provided in Figure 4.3.2. In the scenario using the Default RONR forwarding mechanism, the average latency recorded for the 36 node grid was 618 milliseconds, while with the TM-RONR mechanism, it decreased to 589 milliseconds. In the larger 49 node grid, the Default RONR exhibited an average latency of 763 milliseconds, whereas the TM-RONR showed a reduced latency of 742 milliseconds.

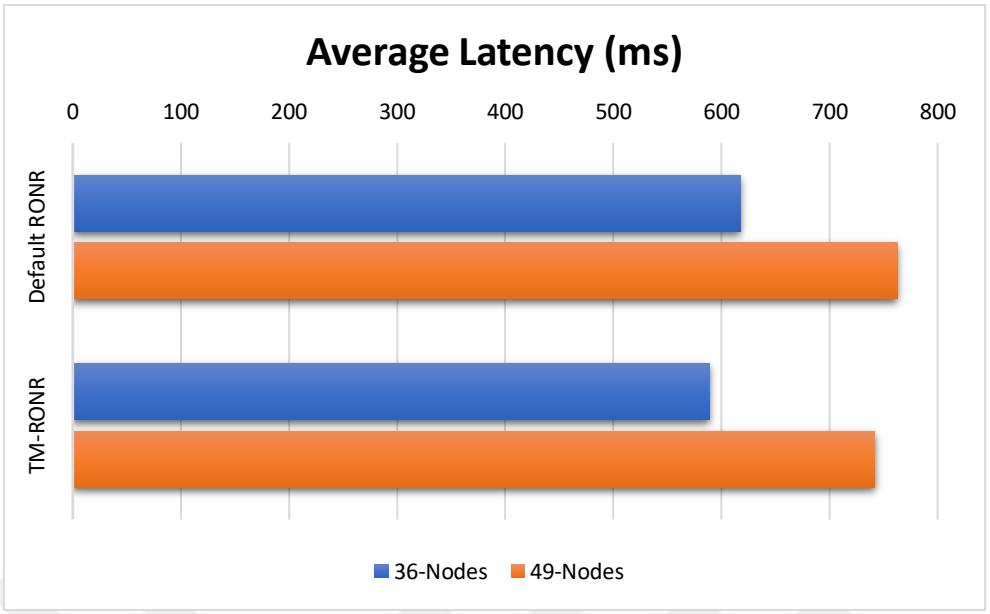


Figure 4.3.2. Average Latency Results with Default RONR Forwarding Mechanism and TM-RONR Forwarding Mechanism under Topologies with 36-nodes and 49-nodes

Analyzing the results reveals a consistent trend where the TM-RONR forwarding mechanism showcased lower average latency in comparison to the Default RONR across both grid topologies. In the 36 node grid, the TM-RONR achieved a notable reduction in latency by 29 milliseconds. Similarly, in the larger 49 node grid, the TM-RONR demonstrated improved performance with a decreased latency of 21 milliseconds compared to the Default RONR.

These findings suggest that the TM-RONR forwarding mechanism tends to offer lower average latency, indicating potentially more efficient forwarding of data within the network compared to the Default RONR, especially in smaller network configurations.

4.3.3. Total Network Traffic

Figure 4.3.3 provides results for the total network traffic metric for both topologies (36 nodes and 49 nodes) and forwarding mechanisms. In the scenario using the Default RONR forwarding mechanism, the total network traffic recorded for the 36-node grid was 20,050 bytes, while with

the TM-RONR mechanism, it slightly decreased to 19,753 bytes. In the larger 49-node grid, the Default RONR exhibited a total network traffic of 35,353 bytes, whereas the TM-RONR showed a reduced traffic of 34,330 bytes.

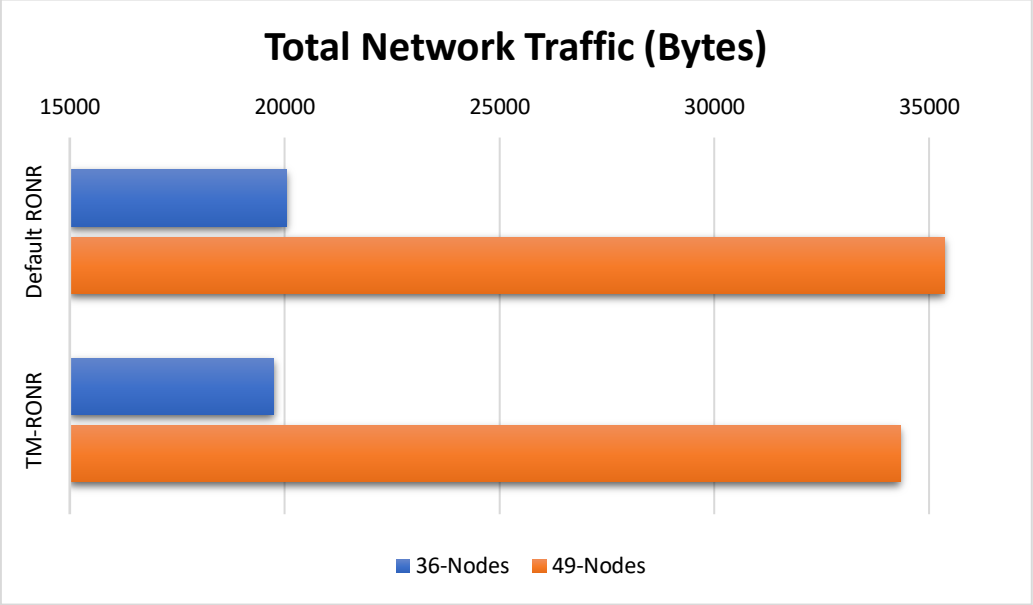


Figure 4.3.3. Total Network Traffic Results with Default RONR Forwarding Mechanism and TM-RONR Forwarding Mechanism under Topologies with 36-nodes and 49-nodes

Analyzing the results reveals a consistent trend where the TM-RONR forwarding mechanism showcased slightly lower total network traffic in comparison to the Default RONR across both grid topologies. In the 36 node grid, the TM-RONR exhibited a reduction of 297 bytes in total network traffic. Similarly, in the larger 49 node grid, the TM-RONR demonstrated decreased traffic of 1,023 bytes compared to the Default RONR.

These findings suggest that the TM-RONR forwarding mechanism might lead to slightly more efficient utilization of network resources, resulting in a reduction in the total network traffic compared to the Default RONR, particularly evident across both network sizes.

4.4. A Content Caching Analysis Study

This section delves into a icontent caching analysis study conducted on the proposed μNDN protocol stack. The section inspects the effects of enabling and disabling the caching mechanism, focusing on critical performance metrics such as success rate, average latency, total interest traffic, and cache hit ratio. Through a methodical comparison of these metrics across different configurations, the study aims to clarify the influence and effectiveness of caching within the μNDN protocol stack. This evaluation evaluates the impact of caching on performance parameters, contributing crucial insights to understanding its role in enhancing the proposed networking framework.

4.4.1. Success Rate

The resulting graph for the success rate metric is provided in Figure 4.4.1. In the scenario where caching was disabled, the success rates across the various node topologies with 16 nodes, 25 nodes, and 36 nodes showed consistently higher values compared to the caching-enabled scenario. The 16 node topology exhibited the highest success rate of 98.53%, followed by the 25 node topology at 87.85%, and finally, the 36 node topology at 84.77%. This trend suggests that, without caching, the success rates were notably higher across all node topologies, indicating a potential dependency of these specific network sizes on direct data retrieval rather than caching mechanisms.

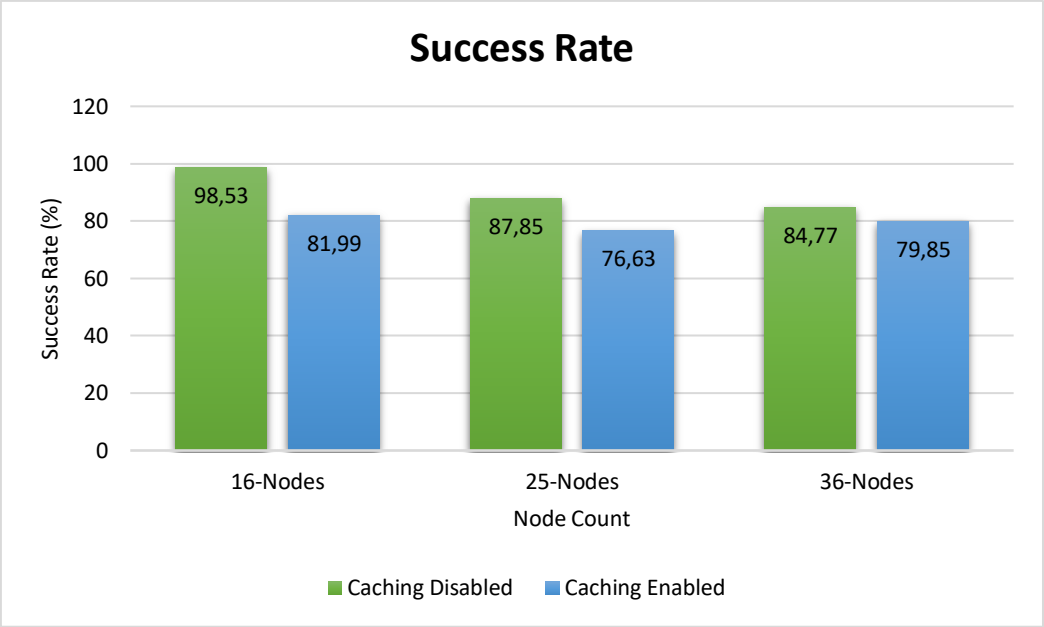


Figure 4.4.1. Success Rate with Caching Enabled/Disabled on µNDN, with 16-nodes, 25-nodes and 36-nodes

Conversely, when caching was enabled, there was a discernible decrease in success rates across all three topologies. The success rates dropped to 81.99% for the 16-node topology, 76.63% for the 25-node topology, and 79.85% for the 36-node topology. This reduction suggests that the introduction of caching mechanisms affected the success rates, indicating potential challenges in the caching process within these network topologies. These values might be higher with more advanced caching policies/mechanisms.

4.4.2. Average Latency

As shown in Figure 4.4.2, average latency results are graphed. In scenarios where caching was disabled, the average latency, measured in milliseconds, showed consistent trends across varying node topologies: 16 nodes, 25 nodes, and 36 nodes. Without caching, the 16 node topology demonstrated the lowest latency at 251 milliseconds, followed by the 25 node topology at 457 milliseconds and the 36 node topology at 613 milliseconds. These findings suggest that smaller network topologies experienced quicker data retrieval, indicating a

potential relationship between network size and latency without the influence of caching mechanisms.

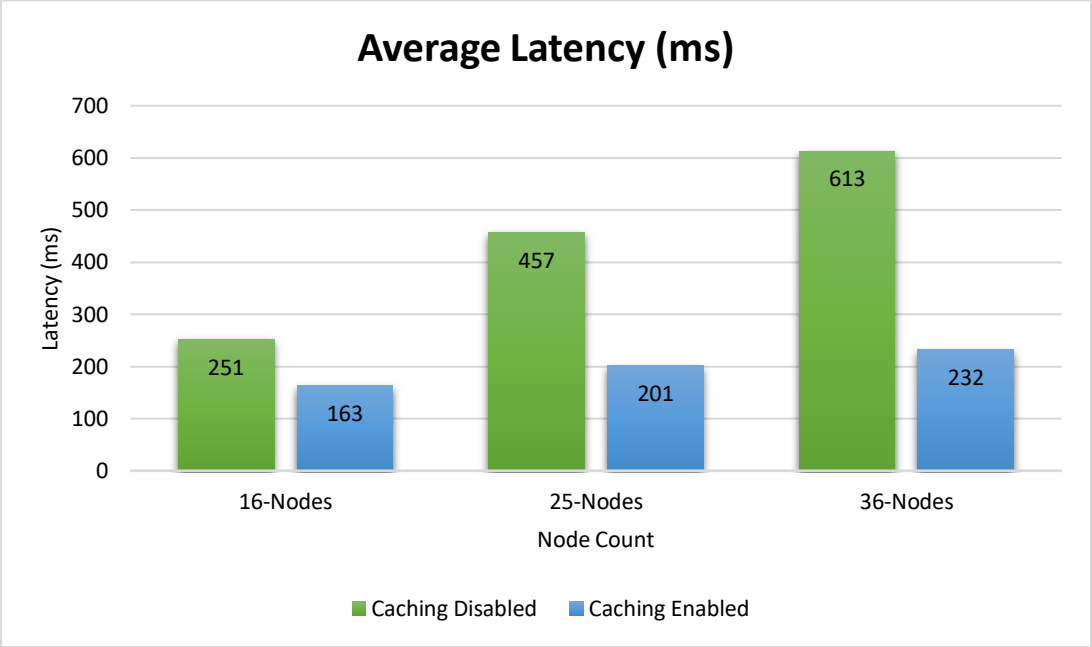


Figure 4.4.2. Average Latency with Caching Enabled/Disabled on µNDN, with 16-nodes, 25-nodes and 36-nodes

Conversely, when caching was enabled, a clear reduction in average latency was observed across all node topologies. The average latency decreased to 163 milliseconds for the 16-node topology, 201 milliseconds for the 25-node topology, and 232 milliseconds for the 36-node topology. This decrease indicates that enabling caching has positively impacted average latency, resulting in improved data retrieval times across all network sizes compared to scenarios without caching.

4.4.3. Total Interest Traffic

As given in Figure 4.4.3, in scenarios where caching was disabled, the total interest traffic, measured in kilobytes (KB), displayed consistent patterns across the different node topologies, 16 nodes, 25 nodes, and 36 nodes. Without caching, the 16 node topology exhibited the lowest

total interest traffic of 1083.4 KB, followed by the 25 node topology at 3084.4 KB and the 36 node topology at 6492.4 KB. These findings indicate that smaller network topologies generated comparatively less total interest traffic, suggesting a relationship between network size and the volume of interest traffic in the absence of caching mechanisms.

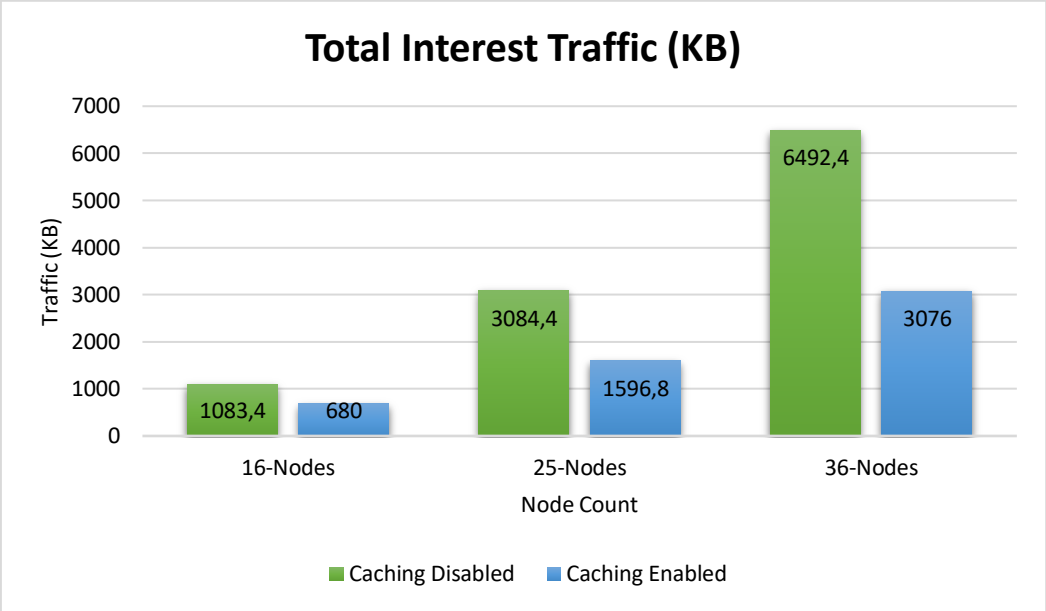


Figure 4.4.3. Total Interest Traffic with Caching Enabled/Disabled on µNDN, with 16-nodes, 25-nodes and 36-nodes

In contrast, with caching enabled, there was a substantial reduction in total interest traffic across all node topologies. The total interest traffic decreased to 680 KB for the 16 node topology, 1596.8 KB for the 25 node topology, and 3076 KB for the 36 node topology. This reduction signifies that enabling caching has notably decreased the total interest traffic generated across all network sizes compared to scenarios without caching.

4.4.4. Total Data Traffic

As we inspect results for the total interest traffic metric, which is given in Figure 4.4.4, without caching, the total data traffic measured in kilobytes (KB) showcased consistent trends across

varying node topologies, 16 nodes, 25 nodes, and 36 nodes. The 16 node topology recorded the lowest total data traffic at 2489.2 KB, followed by the 25 node topology at 6580.6 KB and the 36 node setup at 13704.2 KB. These findings suggest a potential correlation between smaller network topologies and reduced overall data traffic volume in the absence of caching mechanisms.

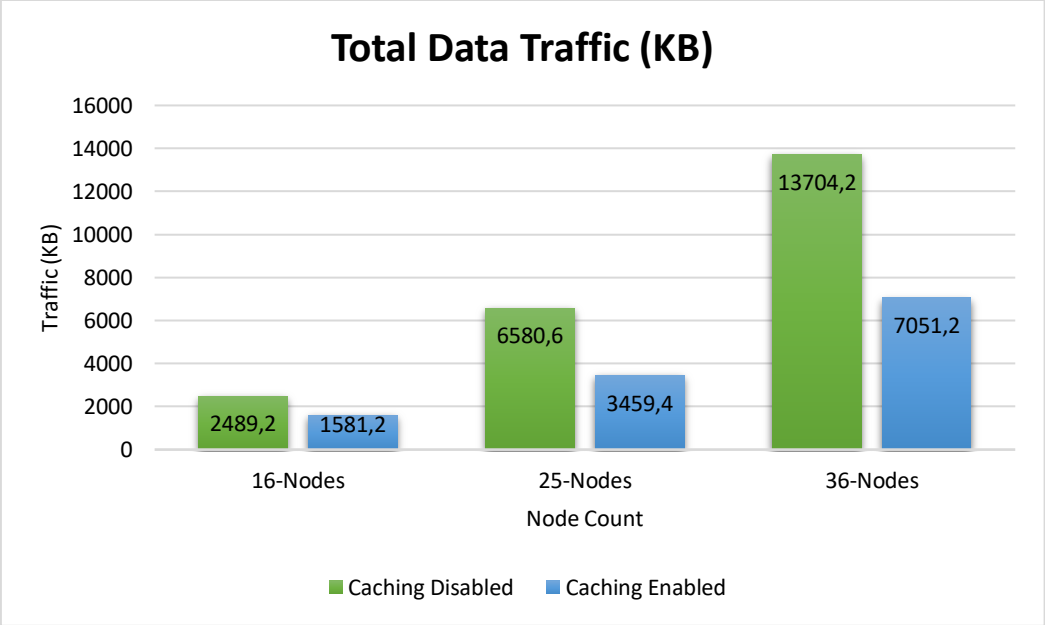


Figure 4.4.4. Total Data Traffic with Caching Enabled/Disabled on µNDN, with 16-nodes, 25-nodes and 36-nodes

However, upon enabling caching, a significant reduction in total data traffic was observed across all node topologies. The total data traffic diminished to 1581.2 KB for the 16 node topology, 3459.4 KB for the 25 node topology, and 7051.2 KB for the 36 node topology. This decline signifies a substantial decrease in the total data traffic across varied network sizes when caching was activated, highlighting the efficiency of caching mechanisms in curbing data traffic compared to scenarios where caching was not employed.

4.4.5. Cache Hit Ratio

The observed cache hit ratio metric result is provided in Figure 4.4.5. Across different topologies, the cache hit ratio showed an ascending trend from the 16 node topology with a ratio of 47.63% to the 25 node topology with 55.58%, and finally to the 36 node topology recording a ratio of 71.45%. These findings suggest an increasing efficiency of the caching mechanism with larger network sizes. The escalating cache hit ratio indicates a higher proportion of successfully retrieved data from the cache, demonstrating the effectiveness of caching in improving data retrieval efficiency as the network size expands. This trend implies that in larger networks, the caching mechanism plays a more substantial role in enhancing data retrieval by efficiently leveraging cached content, resulting in a higher cache-hit ratio.

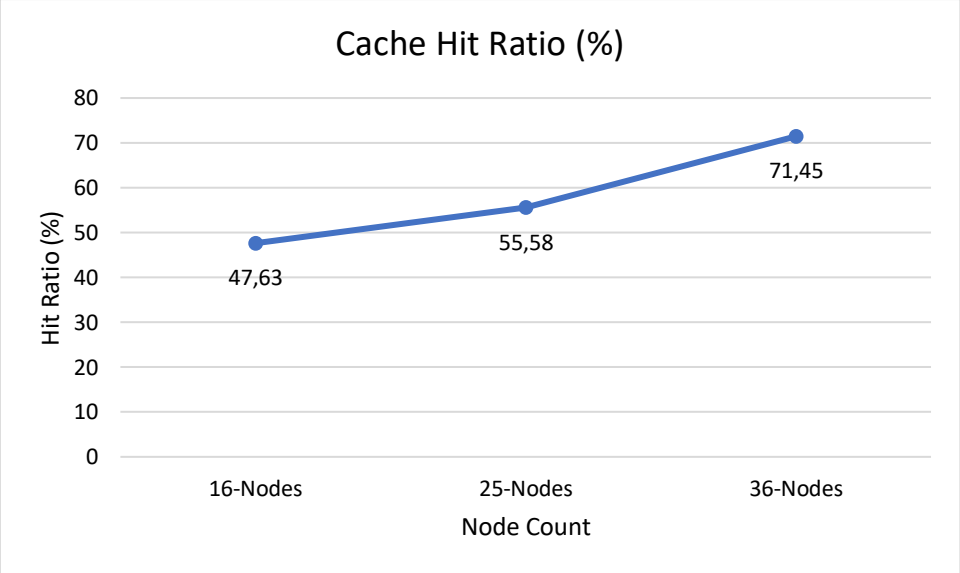


Figure 4.4.5. Cache Hit Ratio when Caching is Enabled on µNDN, with 16-nodes, 25-nodes and 36-nodes

Enabling the caching mechanism revealed contrasting impacts on various performance metrics within the network. Although the success rate exhibited a decrease when caching was enabled, other crucial metrics displayed improvements. Specifically, the average latency, total interest traffic, and total data traffic all demonstrated enhanced outcomes in scenarios where caching

was activated. These improvements suggest that enabling caching can notably optimize data retrieval processes within the network. Such optimization becomes especially advantageous for networks prioritizing low delay and conserving bandwidth. The observed enhancements in average latency and reductions in interest and data traffic volumes indicate that the utilization of caching mechanisms aligns well with the requirements of networks seeking efficient data retrieval, reduced latency, and optimal bandwidth utilization, emphasizing the potential benefits of employing caching in networks with specific performance demands.

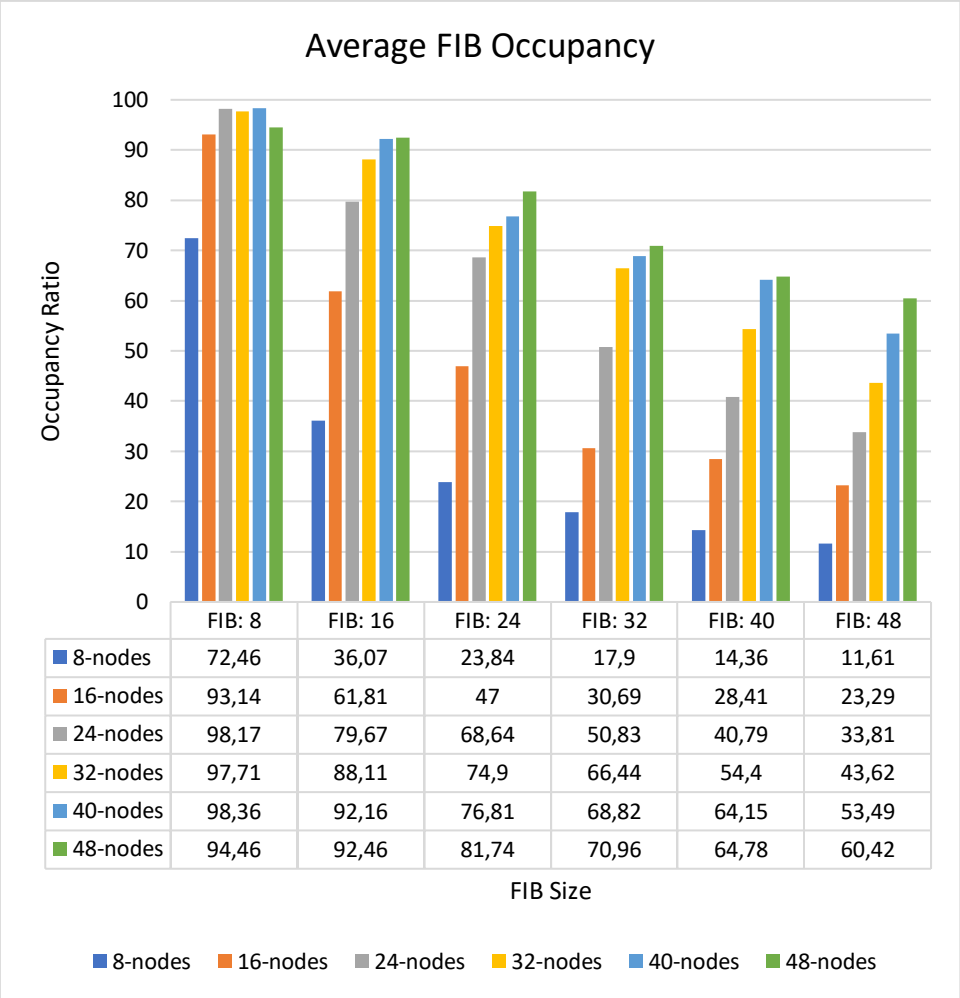
4.5. An FIB Table Size Analysis Study

This section presents the results for performance metrics concerning average FIB occupancy, maximum FIB occupancy, and reliability, highlighting an examination of FIB table sizing. The details of these performance evaluations have been detailed in the preceding sections, offering insights into the network's operational aspects and efficiency through varying FIB table sizes.

4.5.1. Average FIB Occupancy

Figure 4.5.1 illustrates average FIB occupancy percentages across various FIB sizes (8, 16, 24, 32, 40, 48) concerning different node counts (8, 16, 24, 32, 40, 48). This data highlights the average utilization of the FIB space for each combination of FIB size and node count.

At the minimum FIB size of 8, the average occupancy ranges from approximately 72.46% to 98.36% across different node counts. There is a noticeable trend where, as the FIB size increases, the average FIB occupancy consistently decreases. For instance, at the largest FIB size of 48, the average occupancy diminishes, indicating increased FIB capacity and subsequently resulting in lower average occupancy percentages across diverse node counts.



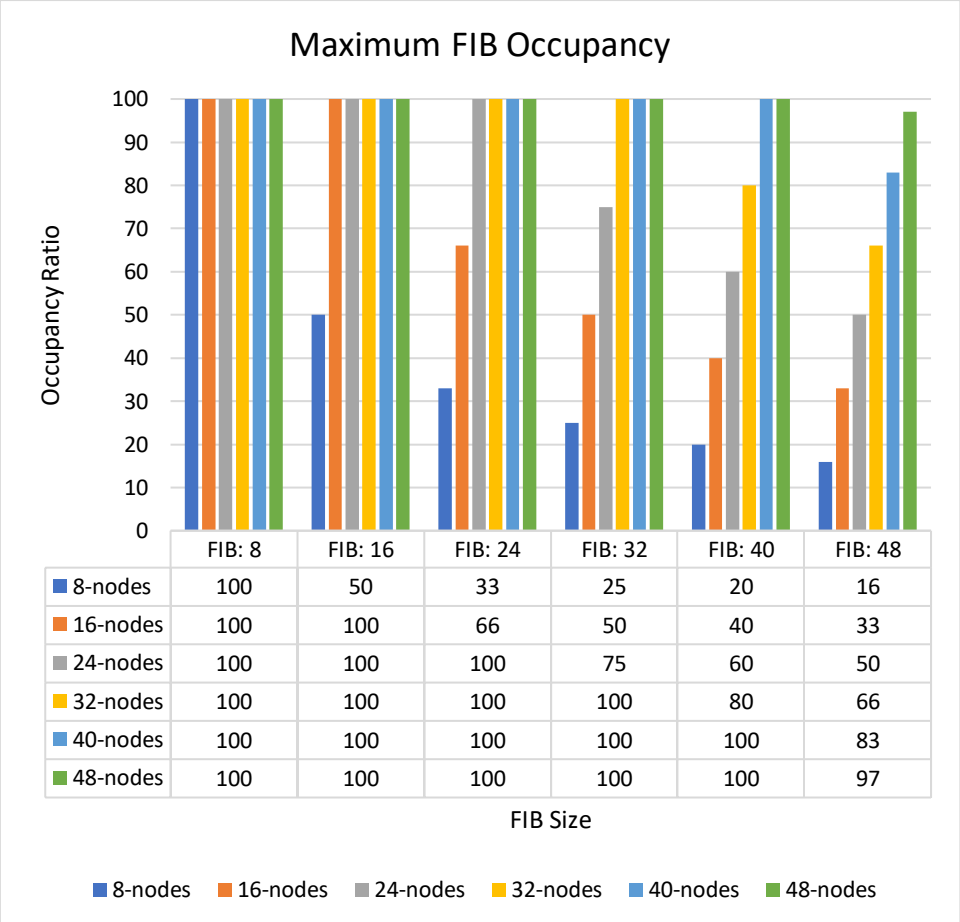
	FIB: 8	FIB: 16	FIB: 24	FIB: 32	FIB: 40	FIB: 48
8-nodes	72,46	36,07	23,84	17,9	14,36	11,61
16-nodes	93,14	61,81	47	30,69	28,41	23,29
24-nodes	98,17	79,67	68,64	50,83	40,79	33,81
32-nodes	97,71	88,11	74,9	66,44	54,4	43,62
40-nodes	98,36	92,16	76,81	68,82	64,15	53,49
48-nodes	94,46	92,46	81,74	70,96	64,78	60,42

Figure 4.5.1. Average FIB Occupancy Results under Different FIB Sizes and Different Network Sizes

Additionally, the data indicates a pattern of increased average FIB occupancy percentages as node counts expand across similar FIB sizes. At FIB size 8, the average occupancy varies across node counts yet generally decreases as the network scales. Conversely, with larger FIB sizes, the impact of node count on average occupancy becomes less pronounced, showcasing a more consistent decrease in average occupancy percentages. This underlines the relationship between FIB size, node count, and average FIB occupancy, suggesting that larger FIB sizes offer greater capacity and consequently result in lower average occupancy percentages across varying network scales.

4.5.2. Maximum FIB Occupancy

Figure 4.5.2 represents the results for the maximum FIB occupancy metric. The data demonstrates that at the smallest FIB size of 8, the maximum occupancy consistently remains at 100% across all node counts. This suggests complete utilization of the FIB space at this specific size without any variation based on the node count. As the FIB size increases from 8 to 48, there is a visible trend wherein the maximum occupancy gradually decreases across different node counts. For instance, at larger FIB sizes, such as 48, the maximum occupancy diminishes, indicating that a larger FIB size accommodates more entries, resulting in reduced FIB utilization percentages across various node counts.



	FIB: 8	FIB: 16	FIB: 24	FIB: 32	FIB: 40	FIB: 48
8-nodes	100	50	33	25	20	16
16-nodes	100	100	66	50	40	33
24-nodes	100	100	100	75	60	50
32-nodes	100	100	100	100	80	66
40-nodes	100	100	100	100	100	83
48-nodes	100	100	100	100	100	97

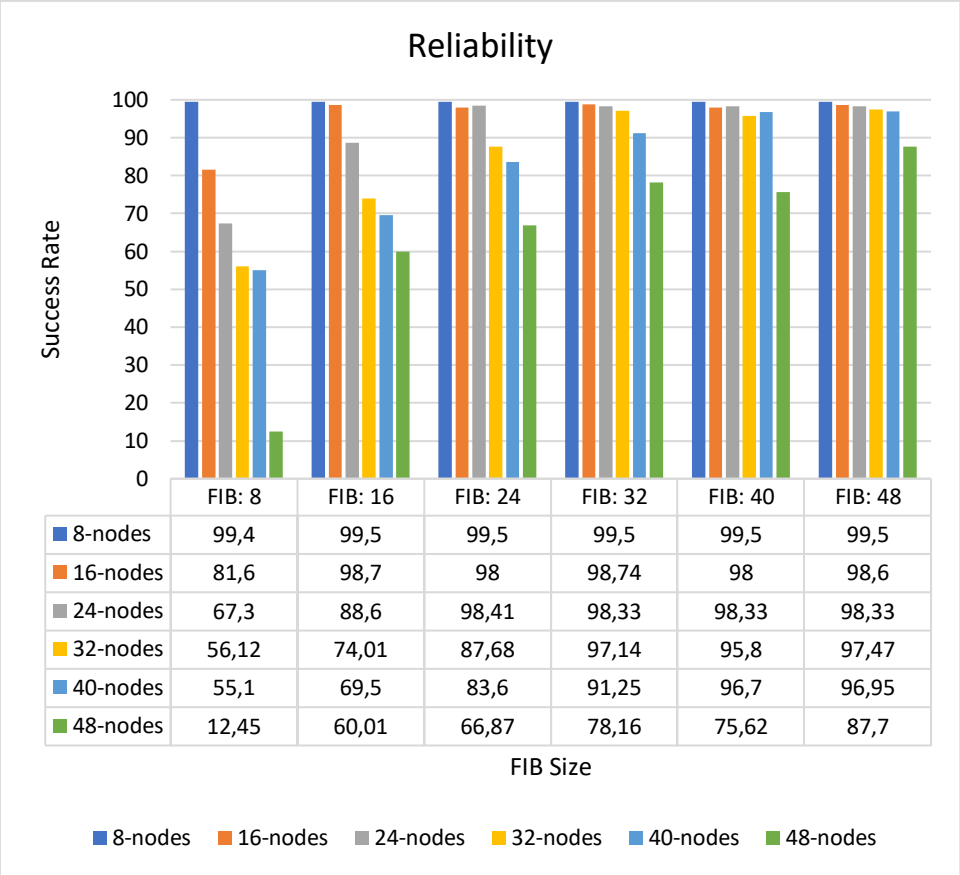
Figure 4.5.2. Maximum FIB Occupancy Results under Different FIB Sizes and Different Network Sizes

Moreover, the data illustrates an upward trend in FIB occupancy percentages as node counts increase across similar FIB sizes. For instance, across the FIB size of 8, the maximum occupancy consistently remains at 100% for all node counts, portraying that irrespective of the network size, the FIB reaches full utilization at this size. This trend signifies a correlation between FIB size, node count, and FIB occupancy, indicating that larger FIB sizes allow for increased capacity and consequently result in lower FIB occupancy percentages across diverse node counts.

4.5.3. Reliability (Success Rate)

Figure 4.5.3 represents the reliability metric across different FIB sizes (8, 16, 24, 32, 40, 48) concerning various node counts (8, 16, 24, 32, 40, 48). Reliability is presented as a percentage, indicating the system's ability to consistently function without failure or errors across different configurations.

At the smallest FIB size of 8, the reliability percentage ranges from 12.45% to 99.4% across different node counts. Notably, as the FIB size increases, there is a general trend of enhanced reliability across diverse node counts. For instance, at the largest FIB size of 48, the reliability substantially improves compared to smaller FIB sizes, showcasing the influence of FIB size on system reliability.



	FIB: 8	FIB: 16	FIB: 24	FIB: 32	FIB: 40	FIB: 48
8-nodes	99,4	99,5	99,5	99,5	99,5	99,5
16-nodes	81,6	98,7	98	98,74	98	98,6
24-nodes	67,3	88,6	98,41	98,33	98,33	98,33
32-nodes	56,12	74,01	87,68	97,14	95,8	97,47
40-nodes	55,1	69,5	83,6	91,25	96,7	96,95
48-nodes	12,45	60,01	66,87	78,16	75,62	87,7

Figure 4.5.3. Reliability (Success Rate) Results under Different FIB Sizes and Different Network Sizes

Moreover, the data showcases fluctuations in reliability percentages concerning node counts across similar FIB sizes. At FIB size 8, the reliability varies considerably across node counts, indicating varied system performance with differing network scales. However, with larger FIB sizes, the impact of node count on reliability becomes less pronounced, showing more consistent reliability percentages across diverse node counts. This underscores the relationship between FIB size, node count, and system reliability, suggesting that larger FIB sizes tend to promote enhanced system reliability across varying network scales.

5. CONCLUSIONS

This thesis presents an experimental exploration of Named Data Networking (NDN), introducing the μNDN protocol stack, introducing the IfNoT mechanism, focusing on key aspects of network performance, caching mechanisms, and FIB table sizing. Two significant contributions, the proposed μNDN protocol stack, and the proposed IfNot mechanism, stand out as pioneering advancements in the realm of NDNoT architectures, underscoring their pivotal roles in enhancing network functionality and attack detection.

The introduction and analysis of the μNDN protocol stack represent a paradigm shift in network architecture within the NDNoT framework. This innovative approach addresses critical challenges while leveraging the inherent advantages of NDNoT. The μNDN protocol stack emerges as a milestone in redefining network infrastructures, promising enhanced efficiency, scalability, and adaptability for future NDNoT-based systems.

Furthermore, the development and implementation of the IfNoT mechanism mark a substantial leap forward in NDNoT's safeguarding and traffic management capabilities. Its integral role in controlling interest traffic and mitigating the impact of malicious attacks reaffirms its significance in ensuring network resilience, stability, and robustness against potential threats. The IfNoT mechanism has demonstrated the capability to enhance the success ratio by up to 28%, reduce the average latency by as much as 31%, and diminish the total interest traffic by up to 58%.

Although this conclusion doesn't constantly highlight the μNDN protocol stack and the IfNoT mechanism, it's important to recognize their significant roles in this study. These pioneering elements redefine the landscape of NDNoT, promising not just incremental improvements but fundamental shifts in how NDNoT-based networks are conceptualized, designed, and secured.

Beyond these pioneering contributions, this study also explores additional issues within NDNoT, such as forwarding mechanism, caching, and FIB sizing. Development and the analysis of the TM-RONR forwarding mechanism, comparing it with the default RONR

forwarding mechanism, focuses on forwarding mechanisms, offering these perspectives on success rates, average latency, and total network traffic, enriching the discourse on NDNoT efficiency.

Another part of this study, a meticulous examination of content caching mechanisms, delineates their impact on success rates, average latency, total interest traffic, and cache hit ratios, elucidating the role and effectiveness of caching within the µNDN protocol stack.

In addition, the in-depth study of FIB sizing unveils correlations between FIB size, node count, and system reliability, elucidating critical relationships influencing network operability and capacity.

While these minor contributions enrich the broader understanding of NDNoT operational intricacies, the µNDN protocol stack, and the IfNot mechanism emerge as major breakthroughs, marking the inception of a new era in NDNoT evolution. Their significance extends beyond incremental improvements, promising fundamental shifts in NDNoT-based network conceptualization, design, and security.

The µNDN protocol stack, as previously mentioned, has been designed to be receptive to advancements. Within the scope of future studies, it is planned to enhance the µNDN protocol stack. Other forwarding mechanisms that currently exist in the literature, apart from VIF and RONR, can be ported to the µNDN protocol stack. Furthermore, the porting of the µNDN protocol stack, which is developed on the Contiki NG operating system, to other operating systems/environments may form the subject of further study.

Interest Flooding Attacks (IFA) for NDNoT environments continue to be a topic of enduring importance. The proposed IfNoT demonstrates quite commendable success in this subject. However, the IfNoT mechanism may not be suitable for every scenario in the domain of NDNoT. Developing an IFA mitigation method enhanced with artificial intelligence techniques, such as machine learning, is also among the planned future studies.

Another study planned for the future involves forwarding mechanisms for NDNoT environments. The research will focus on forwarding mechanisms that can operate efficiently in NDNoT environments with limited devices and potentially lead in the literature. Additionally, adapting existing NDN forwarding mechanisms to NDNoT environments is also included in the plans.

REFERENCES

Aboodi, Ahed, Tat Chee Wan, and Gian Chand Sodhy. 2019a. “Survey on the Incorporation of NDN/CCN in IoT.” *IEEE Access* 7:71827–58. doi: 10.1109/ACCESS.2019.2919534.

Aboodi, Ahed, Tat Chee Wan, and Gian Chand Sodhy. 2019b. “Survey on the Incorporation of NDN/CCN in IoT.” *IEEE Access* 7:71827–58. doi: 10.1109/ACCESS.2019.2919534.

Afanasyev, Alexander, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. 2013. “Interest Flooding Attack and Countermeasures in Named Data Networking.” Pp. 1–9 in *2013 IFIP Networking Conference*.

Agiollo, Andrea, Enkeleda Bardhi, Mauro Conti, Riccardo Lazzeretti, Eleonora Losiouk, and Andrea Omicini. 2023. “GNN4IFA: Interest Flooding Attack Detection With Graph Neural Networks.” *Proceedings - 8th IEEE European Symposium on Security and Privacy, Euro S and P 2023* 615–30. doi: 10.1109/EUROSP57164.2023.00043.

Ahlgren, Bengt, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. 2012a. “A Survey of Information-Centric Networking.” *IEEE Communications Magazine* 50(7):26–36. doi: 10.1109/MCOM.2012.6231276.

Ahlgren, Bengt, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. 2012b. “A Survey of Information-Centric Networking.” *IEEE Communications Magazine* 50(7):26–36. doi: 10.1109/MCOM.2012.6231276.

Alahmri, Bashaer, Saad Al-Ahmadi, and Abdelfettah Belghith. 2021. “Efficient Pooling and Collaborative Cache Management for NDN/IoT Networks.” *IEEE Access* 9:43228–40. doi: 10.1109/ACCESS.2021.3066133.

Al-Share, Rama A., Ahmed S. Shatnawi, and Basheer Al-Duwairi. 2022. “Detecting and Mitigating Collusive Interest Flooding Attacks in Named Data Networking.” *IEEE Access* 10:65996–17. doi: 10.1109/ACCESS.2022.3184304.

Amadeo, Marica, Claudia Campolo, and Antonella Molinaro. 2013. “Enhancing Content-Centric Networking for Vehicular Environments.” *Computer Networks* 57(16):3222–34. doi: 10.1016/J.COMNET.2013.07.005.

Amadeo, Marica, Claudia Campolo, Antonella Molinaro, and Giuseppe Ruggeri. 2014. “Content-Centric Wireless Networking: A Survey.” *Computer Networks* 72:1–13. doi: 10.1016/J.COMNET.2014.07.003.

Amadeo, Marica, Antonella Molinaro, Claudia Campolo, Manolis Sifalakis, and Christian Tschudin. 2014. “Transport Layer Design for Named Data Wireless Networking.” *Proceedings - IEEE INFOCOM* 464–69. doi: 10.1109/INFCOMW.2014.6849276.

Amadeo, Marica, Antonella Molinaro, and Giuseppe Ruggeri. 2013. “E-CHANET: Routing, Forwarding and Transport in Information-Centric Multihop Wireless Networks.” *Computer Communications* 36(7):792–803. doi: 10.1016/J.COMCOM.2013.01.006.

Amadeo, Marica, Giuseppe Ruggeri, Claudia Campolo, Antonella Molinaro, and Giuseppe Mangiullo. 2020. “Caching Popular and Fresh IoT Contents at the Edge via Named Data Networking.” *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2020* 610–15. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162741.

Angius, Fabio, Mario Gerla, and Giovanni Pau. 2012. “BLOOGO: BLOOm Filter Based GOssip Algorithm for Wireless NDN.” *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* 25–30. doi: 10.1145/2248361.2248369.

Anon. n.d. “NSF Future Internet Architecture Project.” Retrieved December 21, 2023 (http://www.nets-fia.net/).

Araujo, Francisco Renato Cavalcante, Andre Luiz Romano Madureira, and Leobino Nascimento Sampaio. 2023. “A Multicriteria-Based Forwarding Strategy for Interest Flooding Mitigation on Named Data Wireless Networking.” *IEEE Transactions on Mobile Computing* 22(12):7000–7013. doi: 10.1109/TMC.2022.3206167.

Baccelli, Emmanuel, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias Wählisch. 2014a. “Information Centric Networking in the IoT: Experiments with NDN in the Wild.” *ICN 2014 - Proceedings of the 1st International Conference on Information-Centric Networking* 77–86. doi: 10.1145/2660129.2660144.

Baccelli, Emmanuel, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias Wählisch. 2014b. “Information Centric Networking in the IoT: Experiments with NDN in the Wild.” doi: 10.1145/2660129.2660144.

Baccelli, Emmanuel, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias Wählisch. 2014c. “Information Centric Networking in the IoT: Experiments with NDN in the Wild.” *ICN 2014 - Proceedings of the 1st International Conference on Information-Centric Networking* 77–86. doi: 10.1145/2660129.2660144.

Baronti, Paolo, Prashant Pillai, Vince W. C. Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu. 2007. “Wireless Sensor Networks: A Survey on the State of the Art and the 802.15.4 and ZigBee Standards.” *Computer Communications* 30(7):1655–95. doi: 10.1016/J.COMCOM.2006.12.020.

Benmoussa, Ahmed, Abdou el Karim Tahari, Nasreddine Lagaa, Abderrahmane Lakas, Farhan Ahmad, Rasheed Hussain, Chaker Abdelaziz Kerrache, and Fatih Kurugollu.

2019. “A Novel Congestion-Aware Interest Flooding Attacks Detection Mechanism in Named Data Networking.” *Proceedings - International Conference on Computer Communications and Networks, ICCCN* 2019-July. doi: 10.1109/ICCCN.2019.8847146.

Bilgili, Sedat, and Alper Kamil Demir. 2022. “A Named Data Networking Stack for Contiki NG OS.” *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2022*. doi: 10.1109/IRASET52964.2022.9738034.

Breivold, Hongyu Pei, and Kristian Sandström. 2015. “Internet of Things for Industrial Automation-Challenges and Technical Solutions.” doi: 10.1109/DSDIS.2015.11.

Carofiglio, Giovanna, Massimo Gallo, and Luca Muscariello. 2012. “Joint Hop-by-Hop and Receiver-Driven Interest Control Protocol for Content-Centric Networks.” *ACM SIGCOMM Computer Communication Review* 42(4):491–96. doi: 10.1145/2377677.2377772.

Carofiglio, Giovanna, Massimo Gallo, Luca Muscariello, and Diego Perino. 2015a. “Pending Interest Table Sizing in Named Data Networking.” *ICN 2015 - Proceedings of the 2nd International Conference on Information-Centric Networking* 49–58. doi: 10.1145/2810156.2810167.

Carofiglio, Giovanna, Massimo Gallo, Luca Muscariello, and Diego Perino. 2015b. “Scalable Mobile Backhauling via Information-Centric Networking.” *IEEE Workshop on Local and Metropolitan Area Networks* 2015-May. doi: 10.1109/LANMAN.2015.7114719.

Chen, Jing, Guanglin Xing, Mengtian Cui, Hong Huo, and Rui Hou. 2019. “Isolation Forest Based Interest Flooding Attack Detection Mechanism in NDN.” *2019 2nd IEEE International Conference on Hot Information-Centric Networking, HotICN 2019* 58–62. doi: 10.1109/HOTICN48464.2019.9063205.

Cheng, Guang, Lixia Zhao, Xiaoyan Hu, Shaoqi Zheng, Hua Wu, and Chengyu Fan. 2020a. “A Network-Wide View-Based Detection and Mitigation of a Sophisticated Interest Flooding Attack.” *Eurasip Journal on Wireless Communications and Networking* 2020(1):1–18. doi: 10.1186/S13638-020-01717-1/FIGURES/5.

Cheng, Guang, Lixia Zhao, Xiaoyan Hu, Shaoqi Zheng, Hua Wu, and Chengyu Fan. 2020b. “A Network-Wide View-Based Detection and Mitigation of a Sophisticated Interest Flooding Attack.” *Eurasip Journal on Wireless Communications and Networking* 2020(1):1–18. doi: 10.1186/S13638-020-01717-1/FIGURES/5.

Compagno, Alberto, Mauro Conti, Paolo Gasti, and Gene Tsudik. 2013. “Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking.” *Proceedings - Conference on Local Computer Networks, LCN* 630–38. doi: 10.1109/LCN.2013.6761300.

Dai, Huichen, Bin Liu, Yan Chen, and Yi Wang. 2012. “On Pending Interest Table in Named Data Networking.” *ANCS 2012 - Proceedings of the 8th ACM/IEEE Symposium on Architectures for Networking and Communications Systems* 211–22. doi: 10.1145/2396556.2396600.

Dai, Huichen, Jianyuan Lu, Yi Wang, and Bin Liu. 2015. “BFAST: Unified and Scalable Index for NDN Forwarding Architecture.” *Proceedings - IEEE INFOCOM* 26:2290–98. doi: 10.1109/INFOCOM.2015.7218616.

Dai, Huichen, Yi Wang, Jindou Fan, and Bin Liu. 2014. “Mitigate DDoS Attacks in NDN by Interest Traceback.” 381–86. doi: 10.1109/INFCOMW.2013.6970722.

Deering, S., and R. Hinden. 1995. “Internet Protocol, Version 6 (IPv6) Specification.” doi: 10.17487/RFC1883.

Demir, Alper Kamil, and Sedat Bilgili. 2022. “The Design and Implementation of an Information Centric Networking Architecture in Contiki NG OS.” *Proceedings - 2022 European Conference on Communication Systems, ECCS 2022* 38–42. doi: 10.1109/ECCS54035.2022.00016.

Deshpande, Ashwini, G. H. Raisoni, Prajakta Pitale, and Sangita Sanap. 2016. “Industrial Automation Using Internet of Things (IOT).” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 5(2).

Din, Ikram Ud, Hamid Asmat, and Mohsen Guizani. 2019. “A Review of Information Centric Network-Based Internet of Things: Communication Architectures, Design Issues, and Research Opportunities.” *Multimedia Tools and Applications* 78(21):30241–56. doi: 10.1007/S11042-018-6943-Z/FIGURES/7.

Djama, Adel, Badis Djamaa, and Mustapha Reda Senouci. 2020. “Information-Centric Networking Solutions for the Internet of Things: A Systematic Mapping Review.” *Computer Communications* 159:37–59. doi: 10.1016/J.COMCOM.2020.05.003.

Djama, Adel, Badis Djamaa, Mustapha Reda Senouci, and Nabil Khemache. 2022. “LAFS: A Learning-Based Adaptive Forwarding Strategy for NDN-Based IoT Networks.” *Annales Des Telecommunications/Annals of Telecommunications* 77(5–6):311–30. doi: 10.1007/s12243-021-00850-2.

Firdhous, Mohamed Fazil Mohamed. 2017. “Information-Centric Networking.” *Encyclopedia of Information Science and Technology, Fourth Edition* 6556–65. doi: 10.4018/978-1-5225-2255-3.CH569.

Gomez, Carles, Joaquim Oller, and Josep Paradells. 2012. “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology.” *Sensors 2012, Vol. 12, Pages 11734-11753* 12(9):11734–53. doi: 10.3390/S120911734.

Gündoğan, Cenk, Peter Kietzmann, Martine Lenders, Hauke Petersen, Thomas C. Schmidt, and Matthias Wählisch. 2018. “NDN, COAP, and MQTT: A Comparative Measurement Study in the IoT.” *ICN 2018 - Proceedings of the 5th ACM Conference on Information-Centric Networking* 13:159–71. doi: 10.1145/3267955.3267967.

Gündoğan, Cenk, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2020. “Information-Centric Networking for the Industrial Internet of Things.” *Wireless Networks and Industrial IoT: Applications, Challenges and Enablers* 171–89. doi: 10.1007/978-3-030-51473-0_9/COVER.

Gupta, Divya, Shalli Rani, Syed Hassan Ahmed, and Rasheed Hussain. 2020. “Caching Policies in NDN-IoT Architecture.” *EAI/Springer Innovations in Communication and Computing* 43–64. doi: 10.1007/978-3-030-38516-3_3/COVER.

Gutierrez, José A., Marco Naeve, Ed Callaway, Monique Bourgeois, Vinay Mitter, and Bob Heile. 2001. “IEEE 802.15.4: A Developing Standard for Low-Power Low-Cost Wireless Personal Area Networks.” *IEEE Network* 15(5):12–19. doi: 10.1109/65.953229.

Hail, Mohamed Ahmed. 2019a. “IoT-NDN: An IoT Architecture via Named Data Netwoking (NDN).” *Proceedings - 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2019* 74–80. doi: 10.1109/ICIAICT.2019.8784859.

Hail, Mohamed Ahmed. 2019b. “IoT-NDN: An IoT Architecture via Named Data Netwoking (NDN).” *Proceedings - 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2019* 74–80. doi: 10.1109/ICIAICT.2019.8784859.

Han, Hailong, Muqing Wu, Qian Hu, and Ning Wang. 2014. “Best Route, Error Broadcast: A Content-Centric Forwarding Protocol for MANETs.” *IEEE Vehicular Technology Conference*. doi: 10.1109/VTCFALL.2014.6965890.

Haxhibeqiri, Jetmir, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. 2018. “A Survey of LoRaWAN for IoT: From Technology to Application.” *Sensors 2018, Vol. 18, Page 3995* 18(11):3995. doi: 10.3390/S18113995.

Herouala, Abdelkader Tayeb, Benameur Ziani, Chaker Abdelaziz Kerrache, Abdou el Karim Tahari, Nasreddine Lagraa, and Spyridon Mastorakis. 2023. “CaDaCa: A New Caching

Strategy in NDN Using Data Categorization." *Multimedia Systems* 29(5):2935–50. doi: 10.1007/S00530-022-00904-Y/FIGURES/14.

Hidouri, Abdelhak, Nasreddine Hajlaoui, Haifa Touati, Mohamed Hadded, and Paul Muhlethaler. 2022. "A Survey on Security Attacks and Intrusion Detection Mechanisms in Named Data Networking." *Computers 2022, Vol. 11, Page 186* 11(12):186. doi: 10.3390/COMPUTERS11120186.

Hou, Rui, Min Han, Jing Chen, Wenbin Hu, Xiaobin Tan, Jiangtao Luo, and Maode Ma. 2019. "Theil-Based Countermeasure against Interest Flooding Attacks for Named Data Networks." *IEEE Network* 33(3):116–21. doi: 10.1109/MNET.2019.1800350.

Hurali, Lalitha Chinmayee M., and Annapurna P. Patil. 2022. "Application Areas of Information-Centric Networking: State-of-the-Art and Challenges." *IEEE Access* 10:122431–46. doi: 10.1109/ACCESS.2022.3223667.

Jacobson, Van, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. 2009. "Networking Named Content." *CoNEXT'09 - Proceedings of the 2009 ACM Conference on Emerging Networking Experiments and Technologies* 1–12. doi: 10.1145/1658939.1658941.

Karrakchou, Ouassim, Nancy Samaan, and Ahmed Karmouch. 2020. "FCTrees: A Front-Coded Family of Compressed Tree-Based FIB Structures for NDN Routers." *IEEE Transactions on Network and Service Management* 17(2):1167–80. doi: 10.1109/TNSM.2020.2969172.

Kim, Jaebeom, Daewook Shin, and Young Bae Ko. 2013. "TOP-CCN: Topology Aware Content Centric Networking for Mobile Ad Hoc Networks." *IEEE International Conference on Networks, ICON*. doi: 10.1109/ICON.2013.6781983.

Kim, Yusung, Younghoon Kim, Jun Bi, and Ikjun Yeom. 2016. "Differentiated Forwarding and Caching in Named-Data Networking." *Journal of Network and Computer Applications* 60:155–69. doi: 10.1016/J.JNCA.2015.09.011.

Kulkarni, Alok, and Sampada Sathe. n.d. "Healthcare Applications of the Internet of Things: A Review."

Kumar, Naveen, Ashutosh Kumar Singh, and Shashank Srivastava. 2017. "Evaluating Machine Learning Algorithms for Detection of Interest Flooding Attack in Named Data Networking." *ACM International Conference Proceeding Series* 299–302. doi: 10.1145/3136825.3136864.

Kumar, Naveen, Ashutosh Kumar Singh, and Shashank Srivastava. 2021. “Feature Selection for Interest Flooding Attack in Named Data Networking.” *International Journal of Computers and Applications* 43(6):537–46. doi: 10.1080/1206212X.2019.1583820.

Kurose, Jim. 2014. “Information-Centric Networking: The Evolution from Circuits to Packets to Content.” *Computer Networks* 66:112–20. doi: 10.1016/J.COMNET.2014.04.002.

Lee, Ren Ting, Yu Beng Leau, Yong Jin Park, and Mohammed Anbar. 2022. “A Survey of Interest Flooding Attack in Named-Data Networking: Taxonomy, Performance and Future Research Challenges.” *IETE Technical Review* 39(5):1027–45. doi: 10.1080/02564602.2021.1957029.

LI, Chengming, Wenjing LIU, and Koji OKAMURA. 2013. “A Greedy Ant Colony Forwarding Algorithm for Named Data Networking.” *Proceedings of the Asia-Pacific Advanced Network* 34(0):17. doi: 10.7125/APAN.34.3.

Li, Shancang, Li Da Xu, and Shanshan Zhao. 2015. “The Internet of Things: A Survey.” *Information Systems Frontiers* 17(2):243–59. doi: 10.1007/S10796-014-9492-7/FIGURES/7.

Liang, Teng, Wei Huang, Xinyu Ma, Weizhe Zhang, Yu Zhang, and Beichuan Zhang. 2023. “PCLive: Bringing Named Data Networking to Internet Livestreaming.” 36–45. doi: 10.1145/3623565.3623711.

Liu, Chunli. 2012. “Intelligent Transportation Based on the Internet of Things.” *2012 2nd International Conference on Consumer Electronics, Communications and Networks, CECNet 2012 - Proceedings* 360–62. doi: 10.1109/CECNET.2012.6201865.

Liu, Liang, Wenzhi Feng, Zhijun Wu, Meng Yue, and Rudan Zhang. 2020. “The Detection Method of Collusive Interest Flooding Attacks Based on Prediction Error in NDN.” *IEEE Access* 8:128005–17. doi: 10.1109/ACCESS.2020.3008723.

Lu, You, Biao Zhou, Lung Chih Tung, Mario Gerla, Ashwin Ramesh, and Lohith Nagaraja. 2013. “Energy-Efficient Content Retrieval in Mobile Cloud.” *MCC 2013 - Proceedings of the 2nd, 2013 ACM SIGCOMM Workshop on Mobile Cloud Computing* 21–26. doi: 10.1145/2491266.2491271.

Marques, Daniel, Carlos Senna, and Miguel Luís. 2022. “Forwarding in Energy-Constrained Wireless Information Centric Networks.” *Sensors* 22(4). doi: 10.3390/s22041438.

Mars, Dorra, Sonia Mettali Gammar, Abdelkader Lahmadi, and Leila Azouz Saidane. 2019. “Using Information Centric Networking in Internet of Things: A Survey.” *Wireless Personal Communications* 105(1):87–103. doi: 10.1007/S11277-018-6104-8/TABLES/1.

Meddeb, Maroua, Amine Dhraief, Abdelfettah Belghith, Thierry Monteil, Khalil Drira, and Hassan Mathkour. 2019. “Least Fresh First Cache Replacement Policy for NDN-Based IoT Networks.” *Pervasive and Mobile Computing* 52:60–70. doi: 10.1016/J.PMCJ.2018.12.002.

Meisel, M. 2011. “BOND: Unifying Mobile Networks with Named Data Freeform Wireless Networks.” *Wirel. Networks*.

Meisel, Michael. n.d. “Listen First, Broadcast Later: Topology-Agnostic Forwarding under High Dynamics.”

Meisel, Michael, Vasileios Pappas, and Lixia Zhang. 2010. “Ad Hoc Networking via Named Data.” *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM* 3–8. doi: 10.1145/1859983.1859986.

Mulligan, Geoff. 2007. “The 6LoWPAN Architecture.” *Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007* 78–82. doi: 10.1145/1278972.1278992.

Mun, Ju Hyoung, and Hyesook Lim. 2019. “On Sharing an FIB Table in Named Data Networking.” *Applied Sciences 2019, Vol. 9, Page 3178* 9(15):3178. doi: 10.3390/APP9153178.

Muto, Takeshi, Kenji Kanai, and Jiro Katto. 2015. “Implementation Evaluation of Proactive Content Caching Using DASH-NDN-JS.” *2015 IEEE Wireless Communications and Networking Conference, WCNC 2015* 2239–44. doi: 10.1109/WCNC.2015.7127815.

Naeem, Muhammad Ali, Tu N. Nguyen, Rashid Ali, Korhan Cengiz, Yahui Meng, and Tahir Khurshaid. 2022. “Hybrid Cache Management in IoT-Based Named Data Networking.” *IEEE Internet of Things Journal* 9(10):7140–50. doi: 10.1109/JIOT.2021.3075317.

Nasir, Nazib Abdun, and Seong Ho Jeong. 2020. “Testbed-Based Performance Evaluation of the Information-Centric Network.” *International Conference on ICT Convergence* 2020-October:166–69. doi: 10.1109/ICTC49870.2020.9289603.

Ngaffo, Armielle Noulapeu, Walid El Ayeb, and Zied Choukair. 2020. “Information-Centric Networking Challenges and Opportunities in Service Discovery: A Survey.” *2020 8th International Conference on Communications and Networking, ComNet2020 - Proceedings*. doi: 10.1109/COMNET47917.2020.9306088.

Ngo, Minh. 2022. *A Study of Routing Protocols for Ad-Hoc Network Based on Named Data Networking*.

Oh, Soon Y., Davide Lau, and Mario Gerla. 2010. “Content Centric Networking in Tactical and Emergency MANETs.” *2010 IFIP Wireless Days, WD 2010*. doi: 10.1109/WD.2010.5657708.

Oikonomou, George, Simon Duquennoy, Atis Elsts, Joakim Eriksson, Yasuyuki Tanaka, and Nicolas Tsiftes. 2022. “The Contiki-NG Open Source Operating System for next Generation IoT Devices.” *SoftwareX* 18:101089. doi: 10.1016/J.SOFTX.2022.101089.

Patel, Palak, Zunnun Narmawala, and Ankit Thakkar. 2019. “A Survey on Intelligent Transportation System Using Internet of Things.” *Advances in Intelligent Systems and Computing* 882:231–40. doi: 10.1007/978-981-13-5953-8_20/COVER.

Patil, Varun, Tianyuan Yu, Xinyu Ma, and Lixia Zhang. 2023. “Decentralized Photo Sharing via Named Data Networking.” 118–20. doi: 10.1145/3623565.3623755.

Paxson, Dr. Vern, Mark Allman, and W. Richard Stevens. 1999. “TCP Congestion Control.” (2581).

Piro, Giuseppe, Luigi Alfredo Grieco, Gennaro Boggia, and Periklis Chatzimisios. 2014. “Information-Centric Networking and Multimedia Services: Present and Future Challenges.” *Transactions on Emerging Telecommunications Technologies* 25(4):392–406. doi: 10.1002/ETT.2741.

Postel, J. 1980. “User Datagram Protocol.” doi: 10.17487/RFC0768.

Postel, J. 1981. “Transmission Control Protocol.” doi: 10.17487/RFC0793.

Pu, Cong, Nathaniel Payne, and Jacqueline Brown. 2019. “Self-Adjusting Share-Based Countermeasure to Interest Flooding Attack in Named Data Networking.” Pp. 142–47 in *2019 international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*.

Qureshi, Adnan Mahmood, Nadeem Anjum, Rao Naveed Bin Rais, Masood Ur-Rehman, and Amir Qayyum. 2021. “Detection of Malicious Consumer Interest Packet with Dynamic Threshold Values.” *PeerJ Computer Science* 7:1–24. doi: 10.7717/PEERJ-CS.435/FIG-12.

Rai, Sandesh, and Dependra Dhakal. 2018. “A Survey on Detection and Mitigation of Interest Flooding Attack in Named Data Networking.” *Advances in Intelligent Systems and Computing* 706:523–31. doi: 10.1007/978-981-10-8237-5_51/COVER.

Ramya, C. Muthu, M. Shanmugaraj, and R. Prabakaran. 2011. “Study on ZigBee Technology.” *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology* 6:297–301. doi: 10.1109/ICECTECH.2011.5942102.

Rehman, Rana Asif, Syed Hassan Ahmed, and Byung Seo Kim. 2017. “OEFS: On-Demand Energy-Based Forwarding Strategy for Named Data Wireless Ad Hoc Networks.” *IEEE Access* 5:6075–86. doi: 10.1109/ACCESS.2017.2684912.

Rehman, Rana Asif, Tran Dinh Hieu, Hong Min Bae, Sung Hoon Mah, and Byung Seo Kim. 2016. “Robust and Efficient Multipath Interest Forwarding for NDN-Based MANETs.” *2016 9th IFIP Wireless and Mobile Networking Conference, WMNC 2016* 187–92. doi: 10.1109/WMNC.2016.7543988.

Rosa, Eduardo Castilho, and Flávio de Oliveira Silva. 2022. “A Review on Recent NDN FIB Implementations for High-Speed Switches.” *Lecture Notes in Networks and Systems* 451 LNNS:288–300. doi: 10.1007/978-3-030-99619-2_28/COVER.

Rose, Karen, Scott Eldridge, and Lyman Chapin. n.d. “The Internet of Things: An Overview Understanding the Issues and Challenges of a More Connected World.”

Rozhnova, Natalya, and Serge Fdida. 2012. “An Effective Hop-by-Hop Interest Shaping Mechanism for CCN Communications.” *Proceedings - IEEE INFOCOM* 322–27. doi: 10.1109/INFCOMW.2012.6193514.

Salah, Hani, Julian Wulfheide, and Thorsten Strufe. 2015. “Coordination Supports Security: A New Defence Mechanism against Interest Flooding in NDN.” *Proceedings - Conference on Local Computer Networks, LCN* 26-29-October-2015:73–81. doi: 10.1109/LCN.2015.7366285.

Saxena, Divya, Vaskar Raychoudhury, Neeraj Suri, Christian Becker, and Jiannong Cao. 2016. “Named Data Networking: A Survey.” *Computer Science Review* 19:15–55. doi: 10.1016/J.COSREV.2016.01.001.

Shelby, Z., K. Hartke, and C. Bormann. 2014. “The Constrained Application Protocol (CoAP).” doi: 10.17487/RFC7252.

Shinde, Anita, and S. M. Chaware. 2018. “Content Centric Networks (CCN): A Survey.” *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018* 595–98. doi: 10.1109/I-SMAC.2018.8653769.

Sivaraman, Vignesh, Dibyajyoti Guha, and Biplab Sikdar. 2020. “Optimal Pending Interest Table Size for ICN with Mobile Producers.” *IEEE/ACM Transactions on Networking* 28(4):1615–28. doi: 10.1109/TNET.2020.2988713.

Soni, Dipa, and Ashwin Makwana. 2017. “A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT).”

Tan, Lu, and Neng Wang. 2010. “Future Internet: The Internet of Things.” *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings* 5. doi: 10.1109/ICACTE.2010.5579543.

Timilsina, Sankalpa, Davide Pesavento, Junxiao Shi, Susmit Shannigrahi, and Lotfi Benmohamed. 2023. “Capture and Analysis of Traffic Traces on a Wide-Area NDN Testbed.” 101–8. doi: 10.1145/3623565.3623707.

Touati, Haifa, Ahmed Aboud, and Brahim Hnich. 2022. “Named Data Networking-Based Communication Model for Internet of Things Using Energy Aware Forwarding Strategy and Smart Sleep Mode.” *Concurrency and Computation: Practice and Experience* 34(3). doi: 10.1002/cpe.6584.

Tzounis, Antonis, Nikolaos Katsoulas, Thomas Bartzanas, and Constantinos Kittas. 2017. “Internet of Things in Agriculture, Recent Advances and Future Challenges.” *Biosystems Engineering* 164:31–48. doi: 10.1016/J.BIOSYSTEMSENG.2017.09.007.

Umeda, Sayaka, Takashi Kamimoto, Yuri Ohata, and Hiroshi Shigeno. 2015. “Interest Flow Control Method Based on User Reputation and Content Name Prefixes in Named Data Networking.” *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015* 1:710–17. doi: 10.1109/TRUSTCOM.2015.438.

Varvello, Matteo, Ivica Rimac, Uichin Lee, Lloyd Greenwald, and Volker Hilt. 2011. “On the Design of Content-Centric MANETs.” *2011 8th International Conference on Wireless On-Demand Network Systems and Services, WONS 2011* 1–8. doi: 10.1109/WONS.2011.5720195.

Verdouw, Cor, Sjaak Wolfert, and Bedir Tekinerdogan. 2016. “Internet of Things in Agriculture.” *CAB Reviews: Perspectives in Agriculture, Veterinary Science, Nutrition and Natural Resources* 11. doi: 10.1079/PAVSNNR201611035.

Wang, Kai, Huachun Zhou, Yajuan Qin, Jia Chen, and Hongke Zhang. 2013. “Decoupling Malicious Interests from Pending Interest Table to Mitigate Interest Flooding Attacks.” *2013 IEEE Globecom Workshops, GC Wkshps 2013* 963–68. doi: 10.1109/GLOCOMW.2013.6825115.

Wang, Kai, Huachun Zhou, Yajuan Qin, and Hongke Zhang. 2014. “Cooperative-Filter: Countering Interest Flooding Attacks in Named Data Networking.” *Soft Computing* 18(9):1803–13. doi: 10.1007/S00500-014-1275-Z/FIGURES/11.

Wang, Lucas, Alexander Afanasyev, Romain Kuntz, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. 2012. “Rapid Traffic Information Dissemination Using Named Data.” *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* 7–12. doi: 10.1145/2248361.2248365.

Wu, Zhijun, Wenzhi Feng, Jin Lei, and Meng Yue. 2021. “I-CIFA: An Improved Collusive Interest Flooding Attack in Named Data Networking.” *Journal of Information Security and Applications* 61:102912. doi: 10.1016/J.JISA.2021.102912.

Xin, Yonghui, Yang Li, Wei Wang, Weiyuan Li, and Xin Chen. 2016. “A Novel Interest Flooding Attacks Detection and Countermeasure Scheme in NDN.” *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*. doi: 10.1109/GLOCOM.2016.7841526.

Xin, Yonghui, Yang Li, Wei Wang, Weiyuan Li, and Xin Chen. 2017. “Detection of Collusive Interest Flooding Attacks in Named Data Networking Using Wavelet Analysis.” *Proceedings - IEEE Military Communications Conference MILCOM* 2017-October:557–62. doi: 10.1109/MILCOM.2017.8170763.

Xue, Haoyue, Yuhong Li, Rahim Rahmani, Theo Kanter, and Xirong Que. 2017. “A Mechanism for Mitigating DoS Attack in ICN-Based Internet of Things.” *ACM International Conference Proceeding Series* 17. doi: 10.1145/3109761.3109787.

Xylomenos, George, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. 2014a. “A Survey of Information-Centric Networking Research.” *IEEE Communications Surveys and Tutorials* 16(2):1024–49. doi: 10.1109/SURV.2013.070813.00063.

Xylomenos, George, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. 2014b. “A Survey of Information-Centric Networking Research.” *IEEE Communications Surveys and Tutorials* 16(2):1024–49. doi: 10.1109/SURV.2013.070813.00063.

Xylomenos, George, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. 2014c. “A Survey of Information-Centric Networking Research.” *IEEE Communications Surveys and Tutorials* 16(2):1024–49. doi: 10.1109/SURV.2013.070813.00063.

YIN, Yuehong, Yan Zeng, Xing Chen, and Yuanjie Fan. 2016. “The Internet of Things in Healthcare: An Overview.” *Journal of Industrial Information Integration* 1:3–13. doi: 10.1016/J.JII.2016.03.004.

Yu, Keping, Suyong Eum, Toshihiko Kurita, Qiaozhi Hua, Takuro Sato, Hidenori Nakazato, Tohru Asami, and Ved P. Kafle. 2019. “Information-Centric Networking: Research and Standardization Status.” *IEEE Access* 7:126164–76. doi: 10.1109/ACCESS.2019.2938586.

Yu, Meiju, Ru Li, Yingqi Liu, and Yingqi Li. 2017. “A Caching Strategy Based on Content Popularity and Router Level for NDN.” *Proceedings of 2017 IEEE 7th International*

Conference on Electronics Information and Emergency Communication, ICEIEC 2017 195–98. doi: 10.1109/ICEIEC.2017.8076542.

Yu, Yu Ting, Raheleh B. Dilmaghani, Seraphin Calo, M. Y. Sanadidi, and Mario Gerla. 2013. "Interest Propagation in Named Data Manets." *2013 International Conference on Computing, Networking and Communications, ICNC 2013* 1118–22. doi: 10.1109/ICCNC.2013.6504249.

Yuan, Haowei, and Patrick Crowley. 2014. "Scalable Pending Interest Table Design: From Principles to Practice." *Proceedings - IEEE INFOCOM* 2049–57. doi: 10.1109/INFOCOM.2014.6848146.

Yue, Meng, Silin Peng, and Wenzhi Feng. 2023. "MF-RF: A Detection Approach Based on Multi-Features and Random Forest Algorithm for Improved Collusive Interest Flooding Attack." *IET Information Security* 17(3):360–76. doi: 10.1049/ISE2.12100.

Yue, Pengfei, Ru Li, and Bin Pang. 2021. "The Random Content Poisoning Attack in NDN." Pp. 853–60 in *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*.

Zhang, Feixiong, Yanyong Zhang, Alex Reznik, Hang Liu, Chen Qian, and Chenren Xu. 2014. "A Transport Protocol for Content-Centric Networking with Explicit Congestion Control." *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. doi: 10.1109/ICCCN.2014.6911765.

Zhang, Lixia, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, K. C. Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. "Named Data Networking." *ACM SIGCOMM Computer Communication Review* 44(3):66–73. doi: 10.1145/2656877.2656887.

Zhang, Lixia, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D. Thornton, Diana K. Smetters, Beichuan Zhang, Gene Tsudik, Dmitri Krioukov, Dan Massey, Christos Papadopoulos, Tarek Abdelzaher, Lan Wang, Patrick Crowley, and Edmund Yeh. 2010. "Named Data Networking (NDN) Project."

Zhang, Qing Yi, Xing Wei Wang, Min Huang, Ke Qin Li, and Sajal K. Das. 2018. "Software Defined Networking Meets Information Centric Networking: A Survey." *IEEE Access* 6:39547–63. doi: 10.1109/ACCESS.2018.2855135.

Zhang, Xin, and Ru Li. 2019. "An ARI-HMM Based Interest Flooding Attack Countermeasure in NDN." *Proceedings of the 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design, CSCWD 2019* 10–15. doi: 10.1109/CSCWD.2019.8791924.

Zhang, Xin, Ru Li, and Wenhan Hou. 2022. “Attention-Based LSTM Model for IFA Detection in Named Data Networking.” doi: 10.1155/2022/1812273.

Zhang, Zhiyi, Edward Lu, Yanbiao Li, Lixia Zhang, Ucla Zhiyi, Tianyuan Yu, Davide Pesavento, Junxiao Shi, and Lott Benmohamed. 2018. “NDNoT: A Framework for Named Data Network of Things.” *Proceedings of the 5th ACM Conference on Information-Centric Networking*. doi: 10.1145/3267955.

Zhao, Lixia, Guang Cheng, Xiaoyan Hu, Hua Wu, Jian Gong, W. Yang, and Chengyu Fan. 2019. “An Insightful Experimental Study of a Sophisticated Interest Flooding Attack in NDN.” *Proceedings of 2018 1st IEEE International Conference on Hot Information-Centric Networking, HotICN 2018* 121–27. doi: 10.1109/HOTICN.2018.8605965.

Zhi, Ting, Ying Liu, Jiushuang Wang, and Hongke Zhang. 2020. “Resist Interest Flooding Attacks via Entropy-SVM and Jensen-Shannon Divergence in Information-Centric Networking.” *IEEE Systems Journal* 14(2):1776–87. doi: 10.1109/JSYST.2019.2939371.