IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 July 2024, accepted 12 August 2024, date of publication 19 August 2024, date of current version 27 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444903

RESEARCH ARTICLE

INF-NDN IoT: An Intelligent Naming and Forwarding in Name Data Networking for Internet of Things

GHULAM MUSA RAZA[1], IHSAN ULLAH[1], MUHAMMAD SALAH UD DIN[1], MUHAMMAD ATIF UR REHMAN[2], AND BYUNG-SEO KIM[1], (Senior Member, IEEE)

[1]Department of Software and Communications Engineering, Hongik University, Sejong-si 30016, South Korea
[2]Department of Computing and Mathematics, Manchester Metropolitan University, M15 6BX Manchester, U.K.

Corresponding author: Byung-Seo Kim (jsnbs@hongik.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by Korean Government (MSIT) under Grant 2022R1A2C1003549, and in part by the 2024 Hongik University Innovation Support Program Fund.

ABSTRACT Internet of things (IoT) has emerged as a quintessential paradigm of communication systems. Current literature introduces notion of a named data network for IoT (NDN-IoT), optimizing IoT communication by employing name-based networking. However, the advancements introduced by this approach are inadequate when dealing with URL-based naming and forwarding. For instance, length and ambiguities in content names are still open challenges. In addition, the intelligent exploration of content names to discern a forwarding clue is a significant research gap. To achieve intelligent communication, understanding the interest name and acquiring a forwarding clue is crucial. Focusing on this gap, an intelligent naming scheme called INF-NDN IoT is proposed that correlates with a forwarding mechanism as well. The proposed INF-NDN IoT improves the NDN naming schemas by utilizing natural language processing (NLP) techniques and selecting supernodes and ordinary nodes in the network. INF-NDN IoT assigns (forwarding clue) semantic tags to content names as well as to supernodes that in turn perform the semantic forwarding. Experimental results have shown that INF-NDN IoT outperformed existing work, and has better results in terms of name length, name memory utilization, interest satisfaction rate, retrieval time, hop count, and energy consumption.

INDEX TERMS Named data network, Internet of Things, natural language processing, supernodes, naming, forwarding, semantic tag.

I. INTRODUCTION

Internet of Things is a vital component of today's information network, which has expanded the horizons of internet services [1]. Recently, many IoT-based solutions have been introduced in various domains [2], [3], [4], such as smart homes, healthcare, the internet of vehicles, and smart campuses. These applications generate huge amounts of data and require efficient communication and computation mechanisms to perform timely decisions. Although IP-based communication has advantages like unbounded data retrieval from IoT devices, it also has limitations such as complex network configuration, security, and mobility, etc.

Current literature suggests that information-centric networking (ICN) provides a more practical solution to IoT issues like efficient communication and ensuring data security between diverse and heterogeneous devices. Among the various ICN solutions, named data networking (NDN) demonstrates the most potential [5] in IoT communication, owing to its ability to dissociate content from location, in-network caching, scalability, flexible forwarding mechanism, and naming features. Utilizing NDN to build IoT networks has become a popular trend in the industry [6]. NDN highlights the availability and access to content by

The associate editor coordinating the review of this manuscript and approving it for publication was Maurizio Casoni.

assigning unique names to each piece of data. NDN content naming eases content caching using content store (CS) implementation, aggregation using pending interest table (PIT), and content forwarding using forwarding information table (FIB). Although NDN offers a range of possibilities that contribute to the availability and accessibility of content, it also brings certain technical difficulties. One of these challenges is the exploration and understanding of content names. In NDN, request routing, data retrieval, interest satisfaction rate, and content discovery all significantly depend on names [7]. NDN name is text data such as /korea/university/hongik.edu/computer-science/documents/students.txt. Text resolution and length in NDN name is the most ignored problem in NDN [8]. The existing state-of-the-art NDN-based schemes mostly utilize hierarchical or URL-based naming conventions in their name-generation process. Authors in [9] explored that memory utilization is higher due to the URL-based names. Moreover, FIB requires scanning tens or hundreds of characters to find the longest-matched prefix, which is time-consuming and resource-intensive.

Studies referenced in [10] emphasized how the text resolution in the NDN name is affecting the network performance. Hierarchical naming doesn't understand the contextual meanings of names. For instance, when a word like "apple" is used in an interest's name, the network interprets it merely as a literal string, disregarding its contextual meaning, such as fruit or technology company. In addition, NDN doesn't remove other name ambiguities or errors such as phonetic similarity, spelling differences, multiple languages, initials, titles, honorifics, and out-of-order components, which can be found in content names. All these naming errors are even more inflated in NDN-IoT. The reason is the length of hierarchical structures like URLs can vary due to unnecessary characters in them. URL-based content names are long, complex, and unrestrained formats. On the contrary, IoT-based names are typically transient, fleeting, new, with diverse priorities, and from various locations.

Another notable research gap in NDN is semantic forwarding. Content names have the potential to support data forwarding in NDN-IoT [11]. Despite this fact, vanilla NDN follows a broadcast mechanism to forward the interest. NDN overlooks the potential of names in providing contextual or semantic clues for forwarding decisions. For instance, an NDN IoT scenario where a student sends an interest "hongik/sports/accessories/available/resources". Currently, the network might process this interest name as a generic text string. However, with semantic understanding, the network could efficiently forward the request directly to the Hongik smart sports environment. Similarly, if a requester requests interest "Prof/Kim/exams-result". Network after semantic awareness, could instantly forward the request to the campus exam department's database. Therefore, it is important to consider name semantics in NDN to fill the research gap of semantic forwarding. In essence, intelligent naming can support semantic forwarding and efficient data retrieval. By obtaining semantic information from content names, NDN routers can forward NDN packets based on their context, improving the efficiency and effectiveness of data dissemination.

Faced with these challenges and research gaps, this article proposes a unique naming and semantical forwarding scheme for NDN IoT. First, we propose a natural language processing (NLP) [12], [13] based naming scheme and semantic tags. A semantic tag is a forwarding hint that is obtained from the NDN name of interest and FIB names of supernodes. Supernodes are a set of resource-rich nodes identified in the network to enable semantic communication. Subsequently, semantic tags are integrated into the namespace design of interest names and assigned to supernodes.This correlation between naming and forwarding aids in establishing a cohesive framework for semantic communication. In addition, selected supernodes have complete knowledge of the network's topology and each supernode stores the same database known as the complete database (CDB), such as [14]. The major contribution of the proposed methodology is focused on naming.

To summarize, the major contributions of this work are:

1) We propose an intelligent naming and forwarding scheme in NDN IoT, called INF-NDN IoT. INF-NDN IoT introduces a unique naming scheme that employs NLP to shorten the length of the content name and effectively resolves ambiguities that may arise in the content names.
2) INF-NDN IoT introduces supernodes to enable a semantic tag-assisted forwarding strategy which leads to semantic content retrieval. To this end, semantic information is extracted from both the Interest/Data packet names and the FIB names of supernodes. Subsequently, corresponding semantic tags are assigned to both the Interest/Data packets and the supernodes. As a result, packets are forwarded to the supernode that possesses the same semantic tag as the packet.
3) Prototype development and extensive experimentation revealed that INF-NDN IoT has optimized the name length and memory, interest satisfaction rate, hop count, retrieval time, and energy consumption.

The remainder of this article is organized as follows. Section II provides some brief background on NDN and state-of-the-art related works. The design description of the INF-NDN IoT is outlined in section III. Section IV specifies the proposed methodology. Section V presents use cases. Section VI explains the simulation study. Finally, section VII concludes our work and highlights the future direction.

II. BACKGROUND AND RELATED WORK

A. NDN ARCHITECTURE

NDN is a state-of-the-art technology that aims to overcome the limitations of the current Internet. It offers a sophisticated architecture where content is identified through hierarchical meaningful names rather than its physical location. NDN has been identified as capable of meeting the requirements of these newly developed Internet applications [15]. NDN routers have the ability to cache data while ensuring adequate bandwidth to meet the users' demands. This makes the technology highly efficient and reliable in managing and delivering content to its users [16]. The NDN router is endowed with three data structures to attain the NDN stateful communication architecture, as presented in Fig. 1.

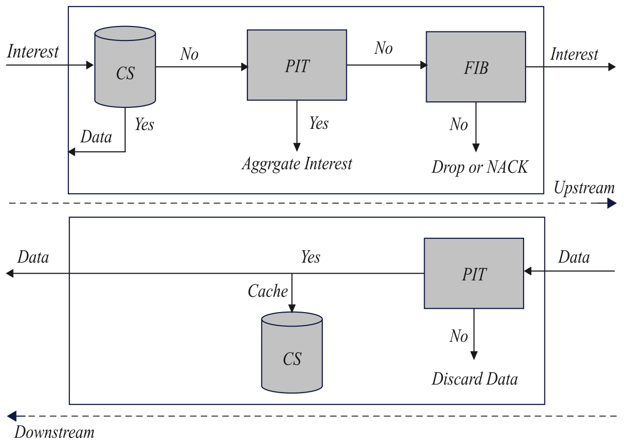


FIGURE 1. **NDN communication process.**

CS behaves like a temporary content store or temporary cache. CS stores the most recent and frequently asked data in it. A PIT is served to store the pending request. If data is not in CS, its name will be investigated in PIT with the incoming interface. If the name is absent in the PIT, it will be forwarded to the FIB and a new entry will be inserted in the PIT. FIB stores the retrieved name announced by the router. It behaves like a control panel to make proper interest forwarding [17], [18], [19].

B. NAMING IN NDN-IOT

Named-based communication relies heavily on naming as it directly influences routing, forwarding, and caching mechanisms. Researchers have come up with various naming schemes for NDN-IoT networks. In this section, the state-of-the-art naming schemes of NDN IoT are summarized and classified into two types, hierarchical and hybrid, as described in Fig. 2. The way interest/data are named in the NDN follows a structure that is arranged in a hierarchy. For instance, a home page created by hongik may have a name like /hongik.my/en/main.html, which is similar to URLs but not necessarily easy for humans to read [20], [21], and also occupy more space in naming due to repeated line slash '/', such are native naming. Name-based service (NBS) like [22] follows the same naming structure and uses command markers to name and access services in a cyber-physical system (CPS) network. These markers are followed by a namespace identifier and then the service names, operation identifiers, and arguments which are delimited by tilde '∼', e.g.

Content centric networking testbed framework (PHINet) [23] implemented naming scheme that allows applications to exchange data over IP using NDN-compliant communication. It uses UDP to support NDN's pull-based mechanism and multitasking. PHINet is implemented using Node.js with a PostgreSQL database for cloud servers and Nod.js with Sqlite3 for the Android operating system to provide native UDP support for client-side applications. NDN for smart home automation systems (NDOMUS) [24] is another naming scheme used in NDN IoT that supports both sensing and management operations in a house. It has two sub-namespace classes: configuration and management, used for network initialization and management, and task (identified by the prefix /task,), used for control and monitoring operations.

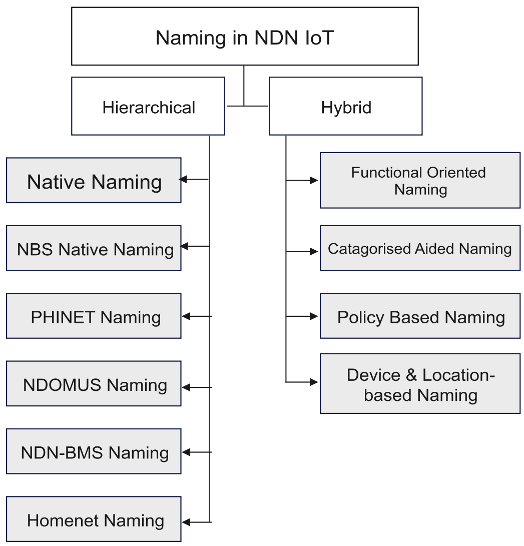


FIGURE 2. **NDN IoT Naming Taxonomy.**

Building management system (BMS-NDN) naming scheme designed by [25] uses a hierarchical structure with a root node that stands for the common prefix for the BMS namespace. They designed the NDN BMS naming where the root node represents the common prefix for the university BMS namespace (/ndn/estc/bms). The building label, located under the root prefix, is divided into floors based on the floor number and further subdivided into rooms based on the room number. Authors in [26] implement the homenet-based naming scheme in the smart home.

Hybrid naming schemes have also been developed by researchers based on specific needs, in which they focus on a particular need and develop the hybrid design accordingly. [27] introduced an entropy-based naming scheme in which the overall name is compressed for efficient name look-up on the machine level. Performance evaluation includes the average name compression ratio (ANCR) and distribution of component compression rate (DCCR). Reference [28] proposed a naming scheme that is based on PURSUIT architecture. The name is divided into their parts. Evaluation includes scheme in the FIB table's name length and look speed. For wireless devices, [29] suggested the multilayer multicomponent hierarchical attribute-value (M2HAV) naming method. The demonstration includes hierarchical-based naming using prefix labeling and a variable-length encoding technique. The naming method is divided into four tiers, with a separate set of characteristics and features used at each level. The name integrated query (NINQ) [30] architecture, includes three components—a hierarchical name, a hash-based flat, and a query-offers a hybrid naming strategy for smart building using NDN. This naming strategy is promising in terms of command satisfaction rate (CSR), interest satisfaction rate (ISR), average delay, energy consumption, and the number of network packets handled, however, naming and query could be too lengthy. The naming of physical IoT objects is driven by using the lightweight named object (LNO) device-based naming scheme introduced in [31]. This scheme can simplify programming and enhance device functionality.

C. NLP INTEGRATION WITH NDN

Content names are text-based strings. Therefore, NLP presents exciting opportunities for efficient communication, leveraging its specialized functions tailored for textual data. Despite garnering researcher attention, there is limited work done in NDN using NLP.

Our target paper, naming and routing scheme for data content objects in information-centric networks (SICN) [32] incorporated the naming scheme with forwarding using NLP. SICN is the only paper to consider semantics in ICN. SICN suggests that publishers and subscribers should hold dynamic addresses that can be changed based on their geography in the network. Also, the names used for dynamic addresses should convey clearly what kind of information they represent. Therefore, the authors suggest a three-dimensional naming system for SICN. User (publisher/subscriber) needs to mention at least one of these three dimensions when they label their data. These three dimensions are the publisher ID, semantic ID, and geographical ID. SICN's proposed forwarding algorithms hold three tables combining the three address dimensions. The first one is the semantic-ID table that connects the semantic address to the publisher ID address. The second one is the geo-ID table that connects the publisher ID address and geographical address, and the third table is the geo-semantic matches semantic address to geographical address. SICN router matches the IDs to forward the interest. If there is no positive match, the router will broadcast the interest to find the best match. On the contrary, if positive matches occur, the router forwards the interest accordingly. However, SICN does not hold a fully semantical forwarding strategy. The reason is that SICN can also forward data if the semantic ID is missing in the routing table. In addition, SICN incorporates geographic ID makes it more suitable for an IP network than NDN because NDN does not consider geographical location and SICN lacks completeness with the constraint of mentioning one dimension by the user.

Another potential work [33] used NLP for naming in an ICN-based weather monitoring system. A monitoring camera captures video or images, and AI technologies on the camera side generate appropriate content names based on the scenes depicted. In contrast, when users provide names in interest packets to retrieve content in ICN, the proposed scheme analyzes priority words from user input sentences on the user side using NLP. Authors classified words contained in sentences to various labels with high probability. However, the authors overlooked the aspect of packet forwarding and semantic communication based on names.

D. SUMMARY OF RELATED WORK

Many scholarly papers have been published with hierarchical and hybrid naming in NDN-IoT. The authors designed schemes according to their appropriate scenarios. However, the improvement is limited by various factors, such as the length of the name, errors in names, and overlooking the potential of names to get forwarding clues. The length of content names becomes critical, as it can impact the lookup process in FIB and occupy more memory. This is particularly crucial in IoT scenarios like data on demand (e.g., military, healthcare, transportation), where minimizing content access latency is essential. To address the name length challenge, further research is necessary to develop naming solutions that reduce name length without imposing arbitrary word limits on users.

Moreover, researchers have ignored the ambiguities and semantical nature of content names. For instance, a requester requests an interest with the name "document/fiction/". However, the forwarder holds a similar-sounding but distinct prefix named "document/friction/". Both documents have different meanings and spellings. Both content names may belong to different smart environments. Because of the longest prefix match algorithm in FIB, the NDN forwards interest towards the interface of "friction" instead of "fiction" since it stops at the first differing character "F". This could lead to a semantic retrieval error since the algorithm prioritizes the longest common prefix without fully considering the requester's intent. Therefore, another notable major loophole in hierarchical and hybrid naming schemes is the lack of exploration of content names. Existing naming schemes have left room for how naming can provide forwarding clues. Table 1 provides an overview of the most significant literature review.

To address these issues, INF-NDN IoT leverages NLP to achieve intelligent naming and forwarding. INF-NDN IoT establishes a correlation between naming and forwarding with the help of semantic tags. The key idea is to explore the FIB names associated with various smart areas within an NDN-based IoT framework. This exploration allows the proposed methodology to identify the primary domain that each smart area handles, such as libraries or classrooms. Subsequently, upon receiving an incoming interest, INF-NDN IoT analyzes its name and directs it to the appropriate smart area that corresponds to its request. For instance, an interest like "/Author/musa-raza/books/friction" would be routed to the smart library within the NDN-IoT framework. In essence, INF-NDN IoT classifies NDN-IoT data into distinct semantic or contextual categories and facilitates packet forwarding based on the semantic category. To our knowledge, we are the first to use NLP to obtain the

TABLE 1. Summary of Literature Review.

Paper	Naming Technique	Research Gap	Major Contribution	Communication Protocol	Semantic Awareness	Major Limitation
[22]	NDN Native	Integration of SOA in WSN IoT	Using URNs instead of URLs to provide a name service-centric architecture.	CCNx	No	Proposed named services require a consistent naming system within the CPS. Missing marshalling and unmarshalling functionality
[25]	NDN Native	Design a new naming and forwarding strategies using ML	Conception of an intelligent HVAC control based on deep learning. Design of a new BMS over NDN. Reduce energy consumption.	Deep learning supported forwarding information base.	No	Overlooks the real-time networking while implementing the DNN.
[23]	NDN Native	Testbed based research gaps between health IoT and NDN	Provide a cheap and easy-to-use testbed for experimentation with Health-IoT over the content-centric network. Architecture that integrates various health sensors such as those worn on the body	Parser supported forwarding strategy	No	Accessibility issue. Not supported with additional communication protocol.
[27]	Hybrid naming	Entropy-based naming	Encoding the original names for efficient communication and look up	Native	No	Lack of update policy for content names. Proposed compactTrie adopts Pointer which is a further waste of memory.
[28]	Hybrid naming (Pursuit based)	Long length of names, difficulty of finding unique content in attribute based naming and naming complexities.	Pursuit based NDN architecture for IoT smart city.	Publish-Subscribe Internet Routing Paradigm (PSIRP) based protocol	No	Too much geographic information is incorporated in name design (location dependent).
[29]	Hybrid naming (M2HAV based)	Naming basic issues, such as globally unique, persistent and secure.	Self-certifying names to achieve a standardized naming scheme	Tree-structured supported categorical based forwarding	No	Name and location is merged in proposed naming.
[32]	Hybrid naming (IDs based)	Naming and routing	Classifying data into the four types and classifying subscriber request into four classes, afterwards naming and routing accordingly	ID based forwarding	Yes (partially)	Broadcasting if IDs are not matched (more delay, more energy, etc) and high length of names.
[33]	Hybrid naming (ML-based)	Lack of design especially supported for the weather monitor system	New naming structure to be used for only weather IoT sensor or monitoring camera applications	NDN native	No	Limited to weather monitor condition.

forwarding clue (semantic tag) from the NDN name and perform semantic forwarding in NDN IoT.

III. DESIGN COMPONENT OF PROPOSED MODEL

The design component of the proposed scheme involves the three types of forwarder nodes to achieve the desired objectives. It is pertinent to accentuate that designated forwarder nodes are resource-rich.

A. PRINCIPAL NODE

Principal node performs NLP-based naming tasks and finds the set of supernodes in the network. The principal node assigns the semantic tags to names and supernodes. Moreover, the principal node is connected to supernodes as a simple tree topology. It is to be noted that the selection of the principal node is out of the scope of this work.

B. SUPERNODE

The interconnection of supernodes ensures that every smart area within the smart campus is encompassed by a corresponding supernode. Supernode holds a shared complete database (discussed in detail in the last of the current section) to assist local FIB. Every supernode is aware of the entire topology including contextual tags of the other supernodes in the network such as [34] and [35]. However, the scalability of supernodes is limited to smart campus.

C. ORDINARY NODES

These nodes are further child nodes of supernodes. One ordinary node can be connected to more than one parent supernode. The ordinary node's FIB contains name prefixes only about parent supernodes.

D. COMPLETE DATABASE (CDB)

CDB is maintained at each supernodes [36]. CDB is similar to link state database (LSDB) in name link state routing (NLSR) [37]. However, in contrast to traditional FIB which contains name prefixes and the corresponding outfaces, the CDB-based FIB includes all the name prefixes served by every supernode in the network. To this end, ordinary nodes and supernodes send link state advertisement (LSA) to nodes in their vicinity, at regular intervals to keep them updated [38].

To understand CDB-based FIB of supernodes, Fig. 3 illustrates the scenario where three supernodes with semantic tag N_1,N_2, and N_3 are producers of prefixes NA-1.1, NA-2.1, and NA-3.1 respectively. At the initial stage, all nodes do not have the prefixes served by each other. As illustrated in Fig. 3 (A), all supernodes exchange LSAs with neighboring supernodes when the network becomes operational. The LSA of each supernode carries prefixes present in its FIB. When N_1 receives LSAs from N_2 and N_3, it updates its CDB-based FIB with the information about N_2's prefix

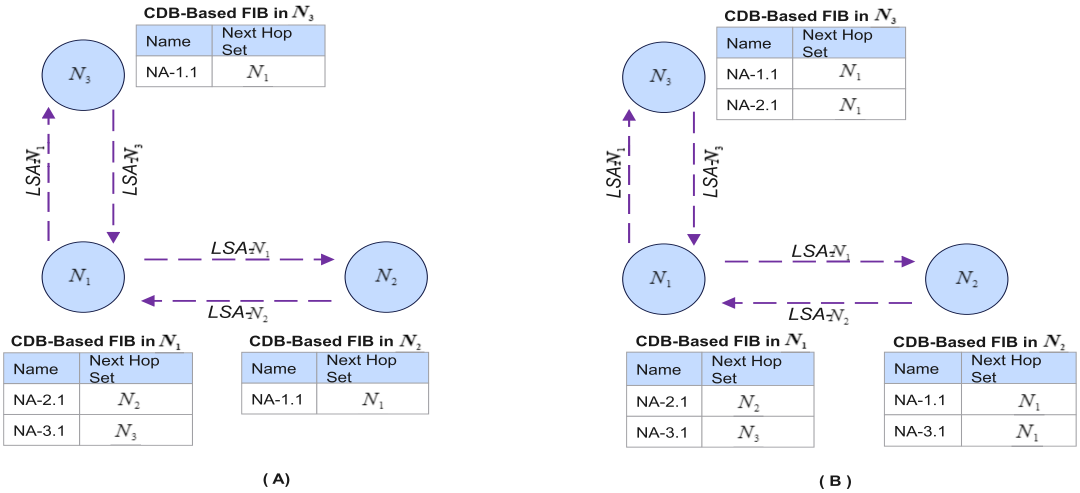


FIGURE 3. Updation of CDB-based FIB.

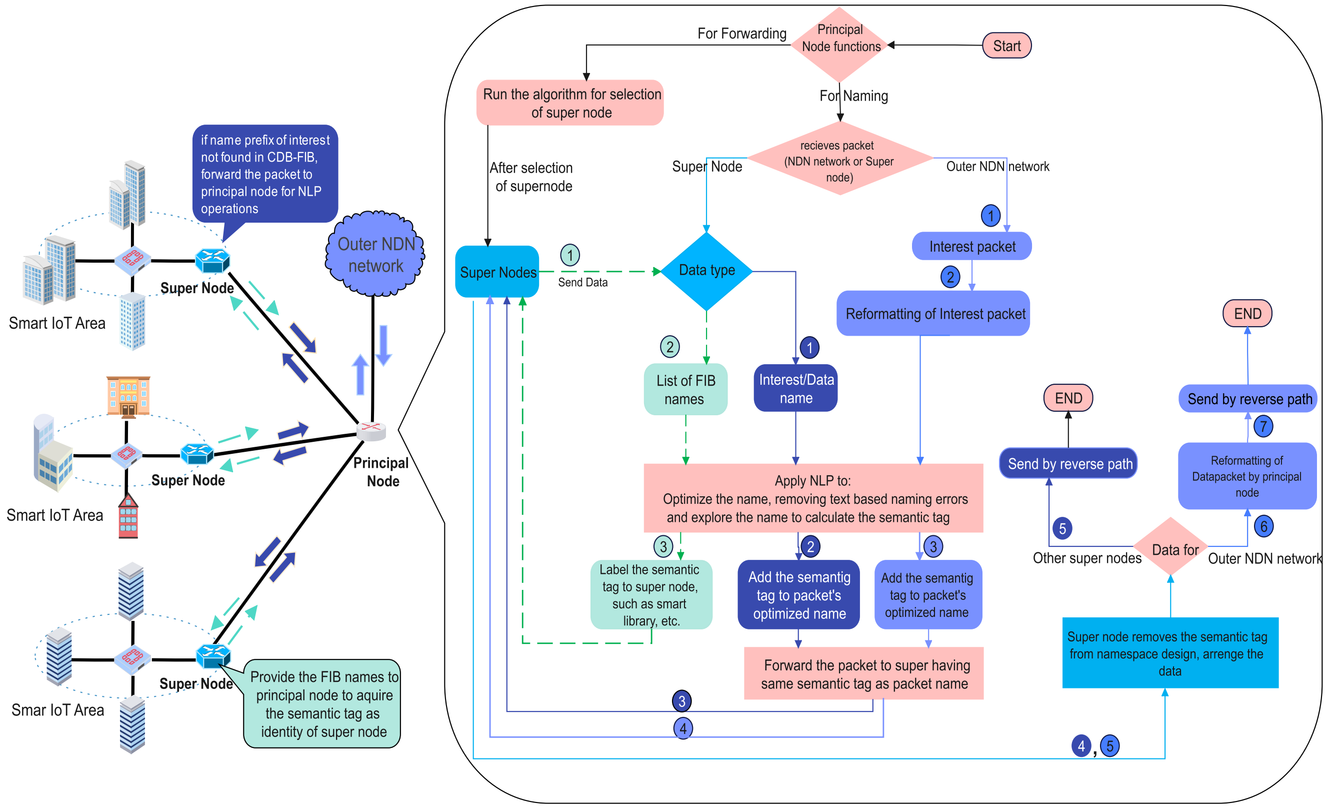


FIGURE 4. INF-NDN IoT architecture.

(NA-2.1), N_3's prefix (NA-3.1) and sets the next hop from which LSA of N_2 and N_3 was received. Same as N_2 and N_3 updates its CDB-based FIB with the information about N_1's prefix (NA-1.1) and sets the next hop. However, N_2 and N_3 are still unaware of prefixes of each other. When all supernodes again exchange LSA, as shown in Fig. 3 (B), N_2 and N_3 receive LSA from N_1, receive prefixes NA-2.1 and NA-3.1 respectively. As prefixes are received from LSA of N_1, as a result, N_2 and N_3 set the next hop N_1 for prefixes NA-2.1 and NA-3.1.

All supernodes exchange LSA at regular intervals. Consequently, any modifications made to the CDB of a single forwarder are automatically disseminated to all other forwarders within the network, ensuring synchronized data across the network. The method for calculating the time required to exchange the LSA is beyond the scope of our work.

IV. PROPOSED MODEL

INF-NDN IoT is deployed on smart campus. The selection of a smart campus for semantic communication is motivated by the diverse IoT configurations present in a smart campus, such as smart labs and smart classrooms, etc. Each smart environment is tailored to manage specific activities; for instance, a smart library handles library-related traffic, and a smart classroom manages classroom traffic. Hence, predicting the destination from the interest name is logical and practical within the smart campus. Figure 4 demonstrates the flow and architectural diagram provided by the INF-NDN IoT.

In this illustration, the principal node receives two types of data from within the smart campus. The first type (FIB names of supernode, briefly discussed in section IV-B3) is represented by a green dotted arrow, presenting the flow of semantic tag assignment to supernodes. The second type is depicted by a dark navy blue arrow, signifying the semantic tag assignment to packets (briefly discussed in section IV-A4). Arrows of sky blue color present data flow from the outer NDN network. For the outer NDN network communication, INF-NDN IoT performs formatting (briefly discussed in section V) after receiving and sending of interest and data packets, respectively. The rationale behind this is outer NDN network's packet namespace designs may be different. Mainly INF-NDN IoT is comprised of two subsections.

1) NLP-based intelligent naming with semantic tag.
2) Semantic tag-assisted supernode-based forwarding.

A comprehensive description of the aforementioned constituents is provided as follows.

A. NLP-BASED INTELLIGENT NAMING WITH SEMANTIC TAGS

INF-NDN IoT develops a namespace design to enable communication between NDN IoT networks in the smart campus.

Each node can forward and receive interest/data to other nodes according to its namespace design. It is to be noted that IDs presented in the namespace designs do not violate the basics of NDN. The rationale for introducing IDs in the namespace is mentioned along with detail. A comprehensive description of the namespaces introduced by the INF-NDN IoT is provided as follows.

1) NAMESPACE DESIGN OF ORDINARY NODE

The namespace design for ordinary nodes, as illustrated in Fig. 5, includes several key components. The first component pertains to the unique ID of the ordinary node. The supernode uses this ID to distinguish the current node from other ordinary nodes and their produced data as well, such as [39], [40], and [41] used the nodes ID in the context of NDN. The INF-NDN IoT does not use the ordinary node ID

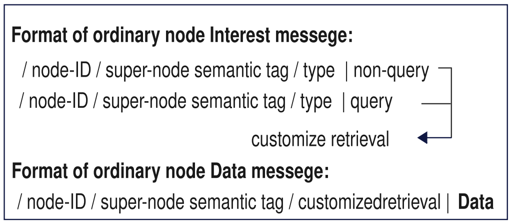


FIGURE 5. Namespace of ordinary node.

for supernode-to-supernode or supernode-to-principal node communication due to the data-oriented architecture of NDN. The second component relates to the supernode semantic tag (section IV-B3), facilitating intersupernode forwarding. The third component represents the type of data generated by the IoT devices (e..g., temperature, pressure, humidity, etc.) or query (section IV-A4) of a random user. This namespace design is used to communicate between the child ordinary node and the parent supernode.

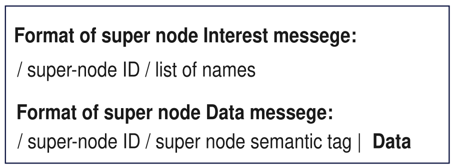


FIGURE 6. Namespace of supernode - A.

2) NAMESPACE DESIGN OF SUPERNODE

To support the roles of supernodes effectively, a supernode requires a well-defined namespace that incorporates the specific type of content produced under its vicinity. INF-NDN IoT uses two distinct namespace designs for supernodes, as illustrated in Fig. 6 and 7. The namespace design presented in Fig. 6, is only leveraged to acquire the semantic tags to supernodes. The first component in the interest packet's namespace denotes a supernode ID, such as [42], [43], and [44] leveraged from nodes ID in NDN.

The second component is a list of content names present in the FIB of supernode. The principal node receives the interest, performs NLP-based operations (section IV-A4), and assigns a semantic tag to each corresponding supernode in exchange.

Once semantic tags are assigned, the rest of the communication is performed by the namespace design of Fig.7. This namespace design serves as the backbone of the smart campus network. Fig. 7 comprises key components of the namespace design when the supernode communicates with other supernodes and principal nodes after acquiring the semantic tags. The first component in Fig.7 is a semantic tag. The second component is the type of data that can be a query or non-query. It is to be noted that semantic tags are not permanently assigned to supernodes. To support the data-oriented architecture of NDN, semantic tags can be changed anytime with contextual variation in network traffic.

Format of super node Interest messege:

/ super-node semantic tag / type | non-query

/ super-node semantic tag / type | query

customize retrieval

Format of super node Data messege:

super node semantic tag / customizedretrieval | **Data**

FIGURE 7. Namespace of supernode - B.

3) NAMESPACE DESIGN OF PRINCIPAL NODE

The namespace design for a principal node includes several components. As illustrated in Fig. 8, the first component can be either a supernode ID or a supernode semantic tag.

Format of principal node Interest messege:

/ super-node ID / type | list of names

/ super-node semantic tag / type | query

customize retrieval

Format of Principal node Data messege:

/ super-node ID / semantic tag | **Data**

/ super-node semantic tag / query | **Data**

FIGURE 8. Namespace of principal node.

The supernode ID is only essential for acquiring semantic tags based on the shared list of FIB names. Conversely, the supernode semantic tag serves as a forwarding clue. The second component is either the list shared by supernodes or the name in the form of the query. The principal node replies with a requested data packet accordingly. The principal node uses its namespace design to only send and receive packets with supernodes. For outer communication, INF-NDN IoT assumes that the principal node only receives interest packets from outside the smart campus network. On receiving interest from outside the smart campus, the principal node applies NLP to change the format of incoming interest according to its namespace and forwards it to appropriate supernodes. After receiving the data from the supernode, the principal node reformats the data packet into the original format (i.e., as per reprieved Interets packet format) and sends it back by reverse path.

4) QUERY OPTIMIZATION AND SEMANTIC TAG TO CONTENT NAMES

In INF-NDN IoT, optimization refers to reducing the length of content names and removing equities from them. There are no restrictions on the word count for queries in INF-NDN IoT, as all forwarder nodes are resource-rich. The workflow and major steps for the query optimization is outlined in Fig. 9 followed by Fig. 10. To clarify the steps involved, INF-NDN IoT focuses on a specific use case: fetching the bus schedule for Hongik University from the Jochiwon station to the Sejong campus. This scenario employs the following namespace for query execution.

$$\begin{aligned}Name = {}&\text{“/students/bustiming?}\\&\text{from\$JOCHIWON\$to+hongik?Sejong}\\&\text{\_campus”}\end{aligned}$$

After performing segmentation, cleaning process, normalization, abbreviation, stemming, and stopword removing steps, the query is optimized as depicted below and does not include extra characters in it, as described.

$$\begin{aligned}Optimized\_name = {}&\text{“studentbustimejochiwon}\\&\text{hongiksejong”}\end{aligned}$$

Afterward, the count vectorizer technique [13] is utilized for the probabilistic formulation of the query. This method transforms the query into vector form, facilitating the subsequent application of NLP algorithms. The semantic score of the optimized query is calculated to understand the context.

To this end, INF-NDN IoT implements the NLP technique called latent dirichlet allocation (LDA) [45], [46] to generate and allocate a semantical tag to each NDN name and supernode. LDA is a probabilistic unsupervised classification technique in topic modeling [47]. Through an iterative process of analyzing the co-occurrence of names in the NDN naming, this semantic model endeavors to identify the contextual tag and represent each document (content names) as a combination of various contextual tags with corresponding weights.

For the implementation of LDA, we define a set of documents as content names and words as the topic or semantic tag, where discrete tag distributions are mapped from LDA. Mathematically, given a vocabulary with N number of tags, document $D = i_1, i_2, i_3, \ldots, i_n$. for each name i in D, we perform the following procedure: Given the joint distribution parameters α and β control tags-names distribution and individual word-to-tag distributions in LDA over the random variables $i_x, j_x, \theta_x, \varphi_T$ to denote the tag-name relationship, are given by:

$$\begin{aligned}P(i_x, j_x, \theta_x, \varphi_T|\alpha, \beta) = {}&P(\theta_x|\alpha)P(\varphi_T|\beta)\\&\prod_{n-1}^{N_x} P(j_{x,n}|\theta_x)P(i_{x,n}|\varphi_{j_{x,n}})\end{aligned} \tag{1}$$

Embedding over θ_x and summing over $j_{x,n}$, the marginal distribution of an NDN name can be computed:

$$\begin{aligned}P(i_x|\alpha, \beta) = {}&\int_{\theta_x}\int_{\varphi_T} P(\theta_x|\alpha)P(\varphi_T|\beta)\\&\left(\prod_{n=1}^{N_x}\sum_{j_{x,n}} P(j_{x,n}|\theta_x)P(i_{x,n}|\varphi_{j_{x,n}})\right)\\&\times d\theta_x\, d\varphi_T\end{aligned} \tag{2}$$

At the last, by computing the marginal probability of each query, the probability of constructing the set of documents is established through the following method:

$$\begin{aligned}P(D|\alpha, \beta) = {}&\prod_{x=1}^{X}\int_{\theta_x}\int_{\varphi_T} P(\theta_x|\alpha)P(\varphi_T|\beta)\\&\left(\prod_{n=1}^{N_x}\sum_{j_x} P(j_{x,n}|\theta_x)P(i_{x,n}|\varphi_{j_{x,n}})\right)\\&\times d\theta_x\, d\varphi_T,\end{aligned} \tag{3}$$

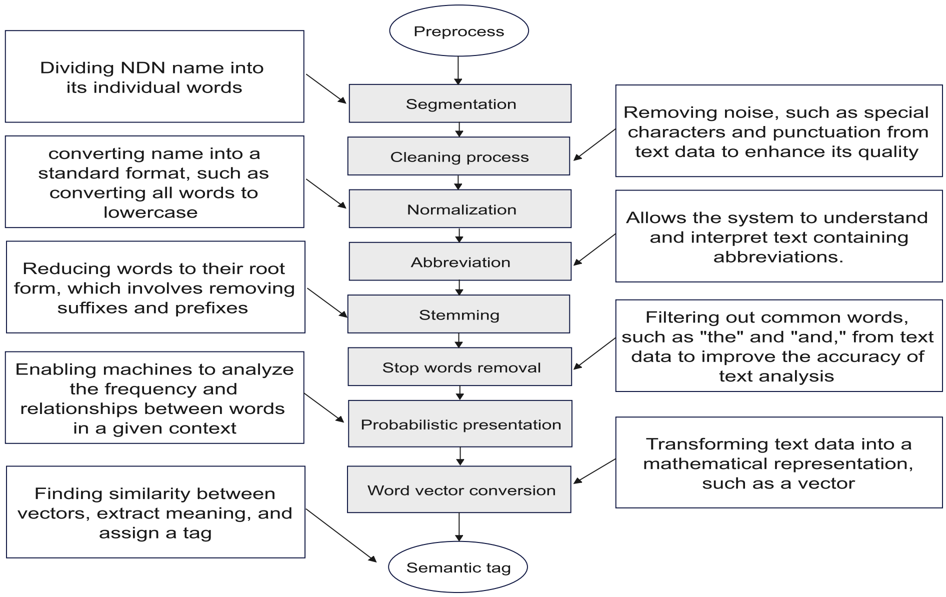


FIGURE 9. NLP-based tasks in intelligent naming.

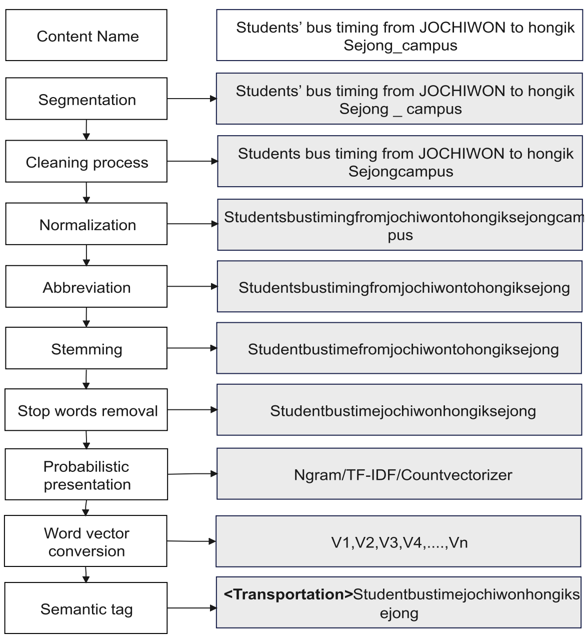


FIGURE 10. An example of finding a tag from the query.

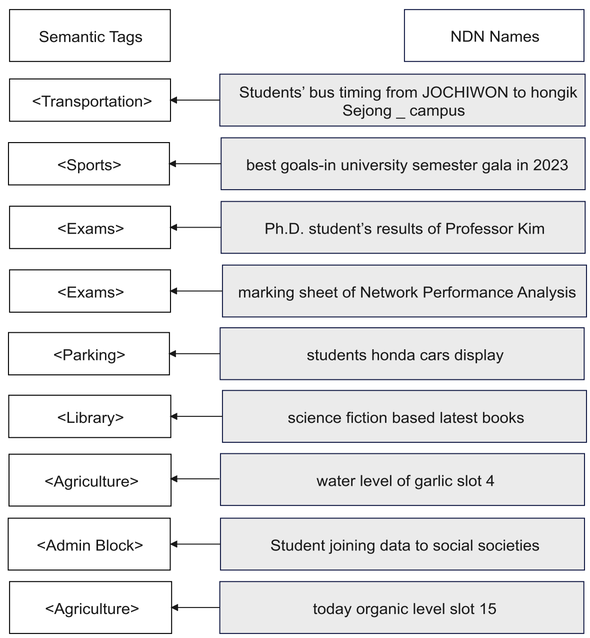


FIGURE 11. Examples of tags corresponding to the content names.

The LDA model is learned using Gibbs sampling, and subsequently, each instance is represented using topic distributions. To assign a new label (contextual tag in our case) to an N_n, we utilize a Gibbs sampler by sampling:

$$P(j_k = T|j_{k-1}, d, i) \propto \frac{n_{it} + \beta}{\sum_v (n_{vt} + \beta_v)} (n_{dt} + \alpha) \qquad (4)$$

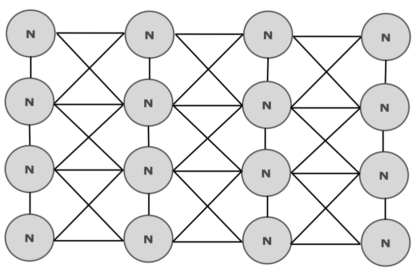
The counts of the contextual or semantic tag with words i or in D are denoted by n_{it} and n_{dt} respectively. The hyperparameters α and β remain the same as before. Following the above-mentioned intelligent naming procedure, every query would be optimized and semantic tags would be assigned to them. The semantic tag summarises the contextual nature of the NDN name into a single word. Along this, semantic tags are assigned to supernodes in the network. The "Transportation" tag has been assigned to the query in the above-said example. This tag describes the forwarding clue for the NDN interest packet, as well (explained in section V). The semantic tags provide a concise representation of the semantical nature of the NDN packets and help to reach their intended destinations or supernodes within the IoT network. As shown in Fig. 11, different content names have been optimized and related semantic tags are assigned. All the intelligent naming operations are implemented on the principal node.

B. SEMANTIC TAG-ASSISTED SUPERNODE-BASED FORWARDING

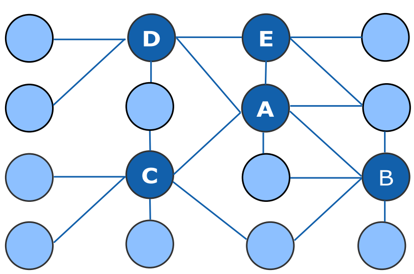
Packet dissemination is accomplished through a semantic tag-assisted supernodes-based forwarding algorithm in the proposed model. Before diving deep into the explanation, we like to explain why we need a set of supernodes in INF-NDN IoT to forward the packets, how does the principal node select and assign the semantic tag supernodes?

1) IDEA BEHIND THE SUPERNODES

The idea behind supernodes is to identify the smallest set capable of maintaining the network's connectivity and ensure its connectivity with the least possible number of connections. To understand the forwarding concept of supernodes, we take a network graph of multiple nodes, as shown in Fig. 12 (A). Upon implementing the selection of the supernodes (algorithm 1), the nodes interconnect each other in such a way that each one establishes communications links with other nodes. This results in a connected subgraph, thereby ensuring comprehensive network connectivity. These specific nodes are referred to as supernodes within the context of INF-NDN IoT. As shown in Fig. 12 (b), the set of supernodes includes nodes A, B, C, D, and E.



(a) Intennectivity of nodes before selection of supernodes.



(b) Intennectivity of nodes after selection of supernodes.

FIGURE 12. Comparison of interconnectivity before and after selection of supernodes.

2) SELECTION OF SUPERNODES

The network initiates randomly for the selection of supernodes in the smart campus. The principal node runs algorithm 1 to find the supernodes in the network.

Algorithm 1 Selection of Supernodes

1: Input: Graph=(V, E), vertex set V and edge set E
2: Source - vertex $\leftarrow s \in V$
3: Auxiliary variables and functions:
4: LSS - list, Covered - list, Neighbours(v) for all $v \in V$
5: Output: LSS - list
6: Initialization: Covered - list = (s), Covered - list = Φ
7: Begin d - selection
8: While |Covered - list| $< |V|$
9: Select a vertex $r \in$ Covered - list and
10: $r \notin$ LSS - list such that r has the maximum
11: neighbors that are not in Covered - list.
12: LSS - list = LSS - list U (r)
13: **for all**$u \in$ Neighbours(r) **and** $U \notin$ Covered - list
14: Covered - list = Covered - list U (u)
15: end while
16: return: LSS - list
17: End d - selection
18: Terminate

The algorithm's execution comprises two fundamental steps: initialization and supernode selection. A detailed description of algorithm 1 is presented as follows: The algorithm operates on an input graph (network topology), Graph=(V, E), where V is the set of vertices (nodes) and E is the set of edges (links) connecting them. The principal node initiates by selecting random "s" as the source vertex line (2). Afterward, the principal node creates two lists: "LSS - list" (linked spanning set) to store supernodes and "Covered-list" to keep track of covered nodes (line 4). The main goal is to find key nodes, called supernodes, that can improve network connectivity.

The algorithm begins with the source node in the "Covered - list" and an empty "LSS-list" (line 6). It enters a loop that continues until all nodes are covered (line 8). In each loop, it chooses a node from the "Covered - list" that hasn't been included in the "LSS - list" and has the most unconnected neighbors (lines 9-11). This node is added to the "LSS - list" as a supernode (line 12). Additionally, the algorithm adds the neighbors of the chosen node to the "Covered - list" if they're not already covered, expanding network coverage (lines 13-15). The loop keeps running until all nodes are covered, ensuring complete network coverage. Finally, the algorithm returns the "LSS - list", which contains the selected supernodes (line 16).

It is to be noted that the updation time of supernodes is application-specific. After every updation, fresh supernodes and corresponding tags would be assigned. Once supernodes are selected, NLP-based algorithms are employed to assign semantic tags to these nodes, as described in the next section. Only supernodes need to update the CDB and send LSA to other supernodes to keep them updated. As a result, communication costs for synchronization are significantly reduced (briefly discussed in section VI-B).

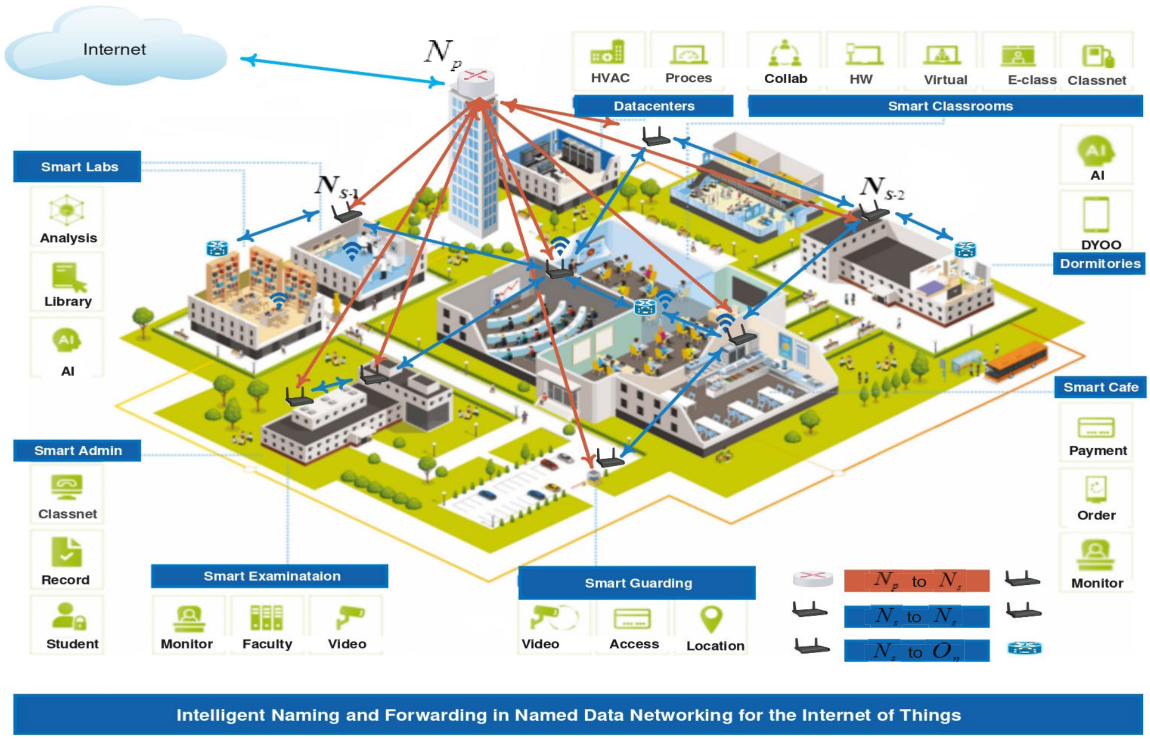


FIGURE 13. Semantic tag-assisted supernodes-based forwarding.

3) SEMANTIC TAGS ASSIGNMENTS TO SUPERNODES

Every supernode initiates traffic generation and forwards a packet to the principal node. This packet comprises a list of FIB entries of the respective supernode, as shown in Fig. 13 where orange arrows represent the principal node N_p to the supernode N_s connectivity and blue arrows represent the supernode to supernode connectivity. The principal node performs the NLP-based LDA to names in the shared list, to understand the contextual nature of traffic in nodes (as briefly discussed in section IV-A.4). As a result, semantic tags are generated and sent back to supernodes in data packet format. In essence, every supernode is enriched with a semantic tag. As a result, the combination of tags and supernodes has categorized the network into different smart areas. This enables NDN packets to be forwarded based on semantic context, thereby potentially optimizing the routing paths. Leveraging this semantic-based forwarding algorithm, the INF-NDN IoT significantly minimizes the latency required to retrieve the data, as compared to issuing multiple interests for each content object. For instance, if the supernode 1 of the $<$ *smartlab* $>$ in smart campus generates the majority of lab-related requests as compared to other nodes. The principal node assigns $<$ *smartlab* $>$ as a semantic tag to that supernode accordingly.

The same applies to the $<$ *smartdormitories* $>$ supernode 2, which handles the network traffic associated with dormitories-related Interest/Data.

However, it's important to note that any supernode is not exclusively limited to the corresponding semantic tag. The principal node can assign multiple tags to single super node.

V. USE CASES SCENARIO

We formulate four distinct use case scenarios to elucidate the concept of tag-assisted supernodes-based forwarding.

- Usecase A: Principal node to supernode communication (query).
- Usecase B: Supernode to supernode communication.
- Usecase C: IoT devices communication (non-query).
- Usecase D: Assigning multiple tags to a single supernode.

These use case scenarios serve as practical illustrations to showcase the effectiveness and applicability of the semantic tag-assisted supernodes-based forwarding approach in IoT scenarios. We also assume push-based communication in INF-NDN IoT, which supports IoT devices to push the data packet into the network without interest, such as [48]. The forwarding strategy for the use cases is outlined in algorithms 2. A comprehensive list of all notations employed in algorithm 2 can be found in Table 2.

A. USECASE A: PRINCIPAL NODE TO SUPERNODE COMMUNICATION

In this case, the principal node receives an interest packet from an external disjoint network source. A stepwise

Algorithm 2 Semantic Tag Assisted Supernodes Based Forwarding

1: Initialize the network randomly;
2: N_p **run algorithm 1 for selection of** N_s;
3: Apply NLP based LDA to N_n **of each** N_s;
4: Assign T **to each** N_s;
5: Initialize the CDB;

6: Use case: A
7: ForP_n **at** N_p
8: Check F_t;
9: If C_n of P_n not found
10: Apply NLP-based text analysis;
11: Optimize the N_n;
12: Perform NLP-based semantic analysis;
13: LDA assign a T to the N_n;
14: Forward the P_n to the relevant N_s;
15: For relevant N_s
16: Removes the T from the C_n;
17: If Packet is P_i
18: Check CS and send the P_d to the reverse path;
19: Else Packet is P_d;
20: Check the PIT and respond correspondingly;
21: EndIf
22: EndFor
23: EndIf
24: Repeat the process for each P_n **at the** N_p ;
25: EndFor
26: Update the CDB;

27: Use case: B,C
28: ForP_n **at the** N_s
29: For each P_i
30: Check the F_t (CDB-based);
31: If not found in F_t **(CDB-based)**
32: Send the packet to N_p;
33: Repeat process from line 7 to 26;
34: Else Found in F_t (CDB-based)
35: Forward the packet to the relevant N_s;
36: EndIf
37: EndFor
38: For Packet is P_d
39: Send the Packet to the N_p;
40: Repeat process from line 7 to 26;
41: EndFor
42: For packet is $P_{s,s}$
43: Check if $P_{s,s}$ **is Interest/Data and responds accordingly**
44: EndFor
45: Repeat the process for each P_n **at the** N_s ;
46: EndFor
47: Update the CDB;

48: Terminate.

TABLE 2. Description of notations.

Notations	Description
P_n	NDN packet
C_n	Content name
F_t	Forwarding table (FIB)
N_p	Principal node
N_s	Supernode
N_o	Ordinary node
T	Semantic or contextual tag
P_i	Interest packet (Query)
P_d	Data packet
CDB	Complete database
$P_{s,s}$	Packet of IoT devices (non-query)

description of interest and data packet exchange shown in Fig. 14 is provided as follows

Step 1: The interest packet arrives at the principal node "P" and carries the name of content to be retrieved, "available/freehostels$for? students_of/ Ph.D".

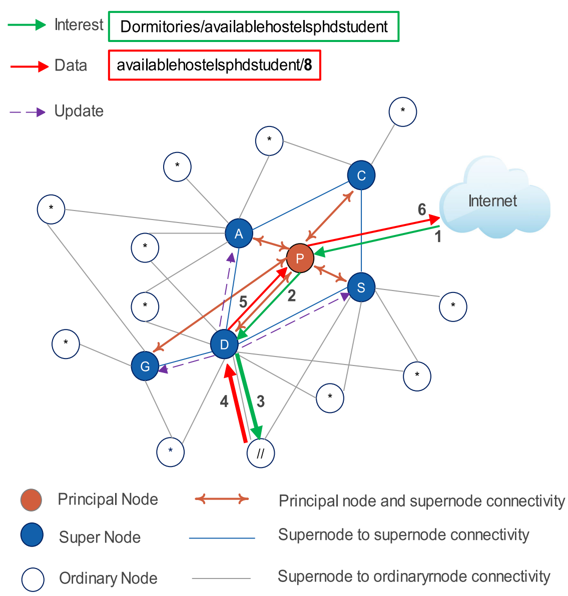


FIGURE 14. Usecase A.

Step 2: Principal node "P" receives the interest. As this interest is received from the outer disjoint NDN network, therefore reformatting of such interest packets is performed. To this end, query optimization is performed and LDA is applied to assign the semantic tag to the optimized query. Afterward, the principal node "P" reformat interest according to the format of INF-NDN IoT, which is "Dormitories | availablehostelphdstudent", where "Dormitories" is semantic tag and "availablehostel-phdstudent" is optimized query. In Fig. 14, NLP-based interest is presented in the green box. At this point, the principal node "P" forwards this interest to the corresponding supernode with the same semantic tag as the interest. To this end, supernode "D" is that corresponding supernode. It is to be noted that if the semantic tag of the interest name does not match with the semantic tags of supernodes, the principal node drops the interest.

Step 3: Supernode "D" receives the interest from the principal node and checks the local cache, if data is not found, then supernode "D" checks the FIB and forwards the interest to its producer child ordinary node whose ID sign is "//". We assume that ordinary nodes are sharing the content names with parent supernodes by LSA (as mentioned in section III-D).

Step 4: Ordinary node "//" receives the interest from supernode "D" and responds to the data packet against requested interest.

Step 5: Supernode "D" receives the data packet from ordinary node "//". It may cache the data into its CS and forward it back to the principal node "P". As all supernodes share the common CDB, supernode "D" sends LSA messages to update the neighbor supernodes about this change in CDB, as presented by purple arrows in Fig. 14.

Step 6: "Dormitories | availablehostels phdstudent | 8" is received from supernode "D" where "8" is the requested content (i.e., available hostels phd students), as presented in the red box in Fig. 14. At this point, the format of the data packet is according to INF-NDN IoT, as presented in the red box. Now, the principal node "P" again converts the format of the data packet into the same format in which its interest was received and sends it back to the internet by reverse path.

It is to be noted that data "8" is included in the namespace to only present the workflow. In simulations, data is not placed within the namespace field of the data packet. The rationale is that the data packet format has a separate field for the placement of data.

B. USECASE B: SUPERNODE TO SUPERNODE COMMUNICATION

In this case, multiple scenarios demonstrate efficient inter-supernode communication within the smart campus IoT network where the supernode may fetch the data from another supernode.

Step 1: As explained in Fig. 15, where the user connected to the child ordinary node whose ID sign is "#" requests an interest packet. The name of interest is "od# / sports | sectionBstrength ", where "#" is the ID of the ordinary node, "sports" is the semantic tag of parent supernode and "sectionBstrength" is the query, as shown in the red box in Fig. 15.

Step 2: Supernode "S" receives the interest. If the requested data in not present in the cache of supernode "S", then supernode "S" checks its CDB-based FIB. If the longest prefix algorithm cannot find the query "sectionBstrength" in CDB-based FIB, supernode "S" forwards interest to principal node "P", repeating steps 2 to 5 from use case A. On the contrary, if the longest prefix matches the name, supernode "S" finds the query

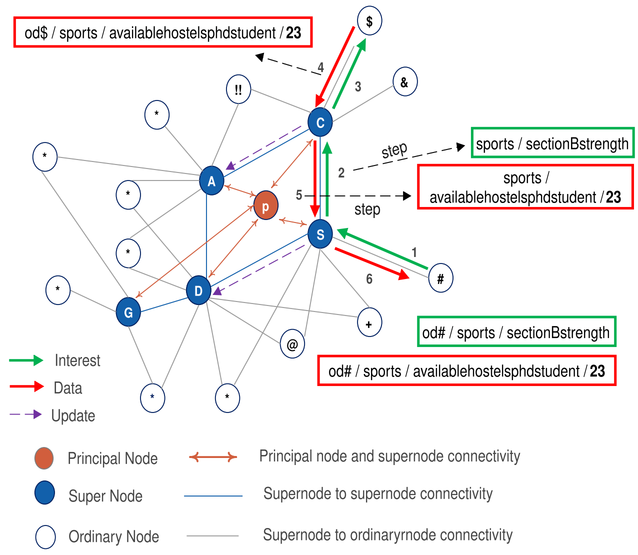


FIGURE 15. Usecase B.

'sectionBstrength" in CDB-based FIB, and data is available on supernode "C'. Supernode "S' removes the ordinary node ID "#" from the format of the packet. It is to be noted that an ordinary node ID is not required for supernode-to-supernode communication. The rationale for removing ordinary node ID is to obey the data-oriented nature of NDN. Supernode "S' forwards the interest to supernode "C'. Supernode "C" receives the interest and checks the local cache.

If requested data is not present in the cache of supernode "C", then supernode "C" forwards the request to producer child ordinary node "$".

Step 3: Ordinary node "$" receives the interest from parent supernode "C" and responds to the data packet against the requested interest.

Step 4: Supernode "C" receives the data packets, caches into CS, and sends LSA to neighbor supernode "A" to keep CDB updated.

Step 5: Supernode "C" removes the ordinary id "$" from the format and forwards the data packets by the reverse path to supernode "S".

Step 6: Supernode "S" caches the data to its CS and forwards the data packet to ordinary node "#", where the requester is attached. Data packet "od# / sports | section strength | 23" is received by ordinary node "#" where 23 is the only presentation of data but placed into another field of the data packet.

C. USECASE C: IOT DEVICES COMMUNICATION

Within a smart campus, an array of IoT devices are usually deployed across various zones, such as the smart library and smart agriculture. Consider a use-case scenario (as shown in Fig. 16) where two distinct sensing devices want to communicate with each other.

Since their communication is solely based on exchanging digital readings, such as temperature and humidity, without

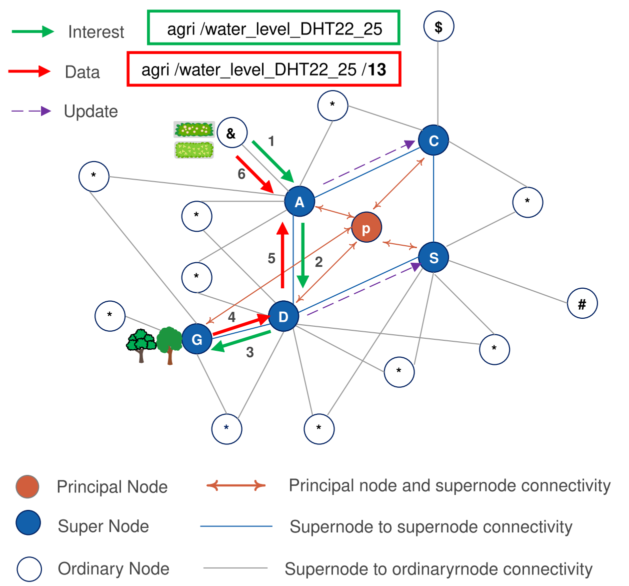


FIGURE 16. IoT devices communication.

any additional context, the generated names are context-free, as presented in Table 3. Therefore, in this instance, non-query-based communication will take place.

TABLE 3. Interpretation of non-query in INF-NDN IoT.

Non-query Names	Description
water.level_-DHT22_25	Select the water level under the sensor DHT22, where values are higher than 25
humi.level_-DHT11_10	Select the humidity level under the sensor DHT1, where values are less than 10
Tem.level_-TP22_10C	Select the temperature under the sensor TP22, where values are higher than 10

A stepwise description of interest and data packet exchange shown in Fig. 15 is provided as follow:

Step 1: As explained in Fig. 16 (smart agriculture area in smart campus), child ordinary node "&" of supernode "A" generates and forwards the interest to parents supernode "A"

Step 2: Supernode "A" receives the interest from ordinary nodes, removes the ordinary node's id "&". In essence, received interest's name is changed to format of supernode to supernode communication, "agri /water_level_DHT22_25" where, "agri" is the semantic tag of parent supernode and "water_level_DHT22_25" shows the type of requested data (non-query) related to wired based sensor "DHT22_25", as presented in the green box in Fig. 16. To this end, supernode "A" does not forward this interest to the principal node "P". Supernode "A" don't have the requested data in the cache and check CDB-based FIB with the help of the longest prefix match algorithm. According to the CDB-based FIB, data against requested content is placed on supernode "G" which is one hop away. Supernode "A" forwards this interest to supernode "D".

Step 3: Supernode "D" receives the interest, also doesn't have the requested data in the cache, and checks the CDB-based FIB. Data against requested content is placed on supernode "G" which is now the neighbor node of supernode "D". Supernode "D" forwards the interest to supernode "G".

Step 4: Supernode "G" receives the interest packet and either checks whether the content has already been generated or senses the environment in real-time. Afterward, supernode "G" stores the data in CS, inserts an entry in CDB-based FIB, and sends back the data to supernode "D".

Step 5: Supernode "D" receives the data, stores it in CS, updates the CDB, and forwards the data to supernode "A". Now, CDB is again updated. Therefore, supernode "D" also sends LSA to update supernode "S" about the change in CDB.

Step 6: Supernode "A" receives the data, stores it in CS, updates CDB, sends LSA to neighbor supernode "C", and forwards the data to child ordinary node "&", as presented in the red box in Fig. 16.

D. USECASE D: ASSIGNING MULTIPLE TAGS TO A SINGLE SUPERNODE

INF-NDN IoT considers random and dynamic traffic generation. However, it is important to note that the location of supernodes, ordinary nodes, and principal nodes remains static. When a consumer/producer switches to a different supernode or ordinary node and initiates communication, the entire network is promptly informed due to CDB. For instance, if students associated with the $<$ *smartclassroom* $>$ tags are switched to $<$ *smartlibrary* $>$ supernode, it would imply that a single supernode is managing traffic for two distinct tags. Therefore, the principal node will add one more tag $<$ *smartclassroom* $>$ to supernode $<$ *smartlibrary* $>$. Now supernode $<$ *smartlibrary* $>$ will deal as dual tag $<$ *smartclassroom_library* $>$. In this situation, all classroom-related traffic would be forwarded to $<$ *smartclassroom_library* $>$.

VI. EXPERIMENTAL ANALYSIS

This section aims to assess the tradeoffs of the INF-NDN IoT system by conducting a comprehensive simulation study. We aim to compare its performance with the state-of-the-art scheme SICN [32] (briefly discussed in section II-C). We begin by describing our simulation setup and subsequently present the results obtained from our simulations.

A. EXPERIMENTAL SETUP

In our simulation setup, we implement INF-NDN IoT in Python using python-NDN [49], pyndn2 [50], and NLTK [51]. Ten different semantic tags are obtained from LDA. LDA is trained over 1500 NDN IoT content names that are relevant to the target supernode. The network model includes a total of sixteen nodes. All the nodes follow wired-based communication model. Five out of sixteen nodes are supernodes converged due to CDB. Four out of sixteen nodes are randomly selected as consumer nodes and two nodes are publishers. The transmission range of ordinary nodes is limited to one hop (i.e., to the supernode they are associated with). The principal node is limited to supernodes in the network(i.e., to the supernodes they are associated with). The principal node is located in the center of the network and is reached by its supernodes. In simulation study, we considered the following evaluation metrics.

1) Name Length: Total characters in the names of interest/data packets.
2) Name Memory Utilization: Memory utilization by names of interest/data packet.
3) Retrieval Time: Total amount of time the interest packet takes to successfully reach the producer, time taken by the producer to process, and Data packet to reach back to the requester.
4) Number of Hops: It refers to the count of intermediary nodes that packets traverse between the provider and consumer.
5) Intrest Satisfaction Rate: Total number of Data packets successfully retrieved to interest packets in the network.
6) Query Satisfaction Rate: Ratio of total successfully satisfied data against a total of query-based interest sent.
7) Energy Consumption: The aggregate energy is utilized by the nodes for the transmission of the interest packet.

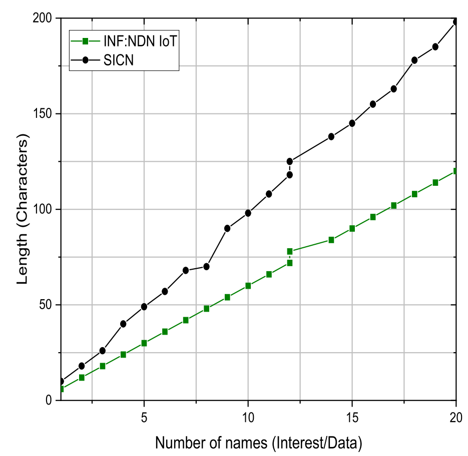


FIGURE 17. Content name length.

B. EXPERIMENTAL RESULTS

1) NAME LENGTH

The average name length as a function of a number of content names (interest and data) can be visualized in Fig. 17. Results demonstrate that the average length of content names is decreased by approximately 40.05 % compared to SICN. The rationale is that SICN does not incorporate any mechanisms to shorten the length of content names. In contrast, INF-NDN IoT effectively reduces the character count of content names by leveraging various NLP techniques such as data cleaning, normalization, abbreviation, stemming, stopword removal, and segmentation.

Our findings demonstrate the effectiveness of incorporating NLP techniques in streamlining content names, resulting in more concise names. It is to be noted that name lengths are related to entries of content names in CDB-based FIB. The name length does not include the semantic tag. The rationale behind this is that INF-NDN IoT removes the tag while creating an entry in the CDB-based FIB.

2) NAME MEMORY UTILIZATION

Memory utilization is directly proportional to the length of names. A larger length of NDN name increases the memory footprints. Fig. 18 demonstrates the relationship between memory utilization (in megabytes) of content names and the total number of content names with respect to INF-NDN IoT and SICN. Results show that INF-NDN IoT improves memory utilization by harnessing the capabilities of NLP, resulting in shorter content names. INF-NDN IoT does not introduce extra components or information in the structure of the NDN name. On the contrary, SICN incorporates additional identifiers, increasing its overall length and memory footprint. In addition, SICN also includes longer and more descriptive names and repetitions of "/" or "$". These extra characters increase the overall memory utilization.

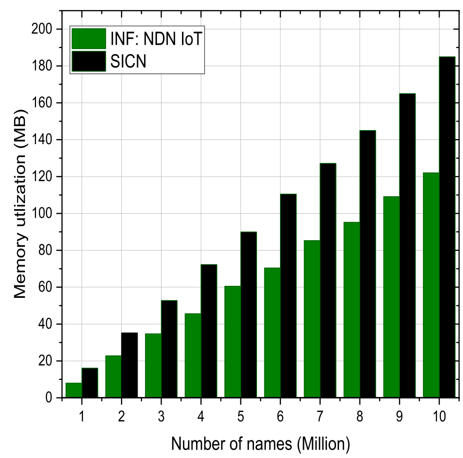


FIGURE 18. Content name memory.

3) NUMBER OF HOPS

Fig. 19 shows the average no of hops count on the y-axis as a function of the number of nodes on the x-axis. Results demonstrate that the INF-NDN IoT reduces the number of hops required to reach the destination. One core reason for the reduction in the number of hops is that all supernodes are assigned semantic tags and share CDB. Due to the semantic tags, supernodes perform semantic and selective forwarding. Semantic-based forwarding enables supernodes to select interfaces aligning with the specific semantic requirements of the transmitted data.

As a result, not all nodes participate in packet forwarding. By involving fewer nodes in the forwarding process, the number of hops a packet needs to traverse is 4 when overall

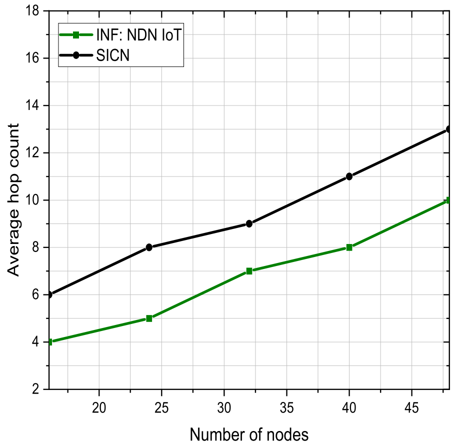


FIGURE 19. Average number of hop counts.

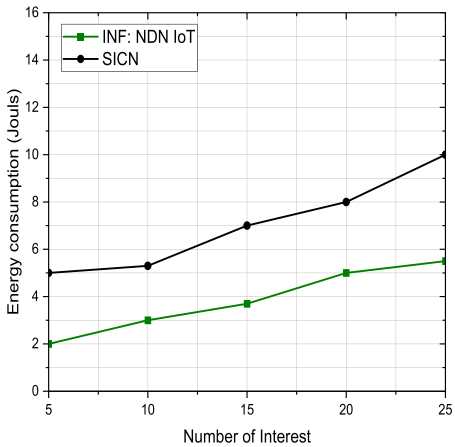


FIGURE 20. Interest energy consumption.

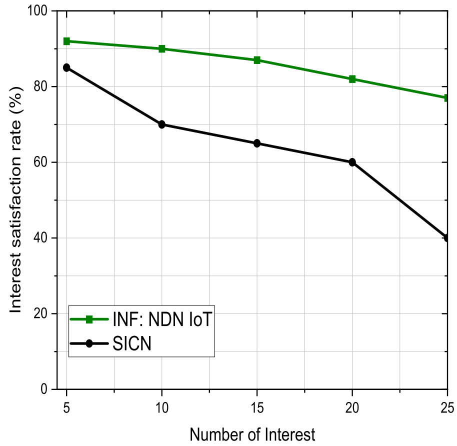


FIGURE 21. Interest satisfaction rate.

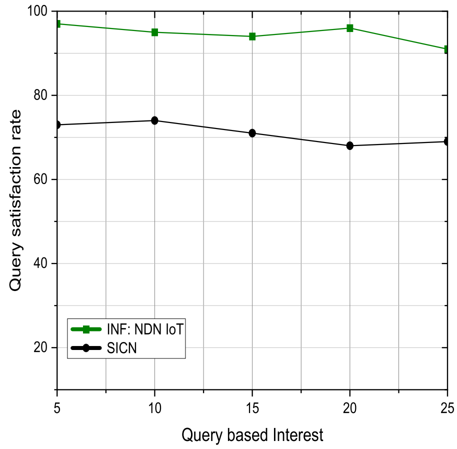


FIGURE 22. Query satisfaction rate.

network nodes are 16. If numbers of nodes are increased to 50, the average hop count for INF-NDN IoT is 10. On the contrary, SICN's average hop count is 7 and 15 when a total number of nodes is 16 and 50 respectively, which is more than INF-NDN IoT. SICN forwards data on behalf of matching IDs, namely, geo, semantic, and publisher IDs.

In the scenario of mismatching IDs or mobility of publisher and consumer, SICN performs broadcasting. As a result, the average hop count is increased.

4) ENERGY CONSUMPTION

Total energy consumption as a function of the number of interest packets is illustrated in Fig. 20. INF-NDN IoT consumes less energy than SICN in the case of forwarding interest packets. The reason is that the principal node forwards the interest packet to the producer supernode. Same as, all ordinary nodes directly communicate with supernodes, eliminating the need for any intermediate node. In the worst case, if the supernode doesn't have data, supernodes directly forward the interest packet to the producer supernode due to CDB. This also minimizes the intermediate nodes. As a result, INF-NDN IoT prevents the occurrence of broadcast storms and minimizes collisions in the network. Consequently, the overall energy consumption across the network is significantly reduced. In contrast, SICN uses a hybrid approach. It efficiently handles interest flooding by matching three-dimensional IDs. If there's a match, forwarding occurs smoothly; otherwise, SICN broadcasts the interest when IDs don't match. As a result, the energy consumption of SICN is higher than INF-NDN IoT.

5) INTEREST SATISFACTION RATE

Fig. 21 shows the relationship between interest packets and ISR. we increase interest packets from 0 to 25 per second. Results show that INF-NDN IoT performs better. The reason is that INF-NDN IoT directly fetches the content from the producer. The principal node directly forwards the interest packet to the relevant producer or supernode and fetches the data without involving intermediate nodes. This semantic awareness leads to a higher interest satisfaction rate as it improves the precision and accuracy of data retrieval. On the contrary, SICN does not explicitly incorporate semantic tags in the NDN name. It primarily focuses on publisher ID and geographical information and broadcasts in the worst case, thus decreasing the interest satisfaction rate. SICN schemes involve multiple intermediate forwarders in the communication, causing congestion and collisions in the network and thus decreasing ISR.

6) QUERY SATISFACTION RATE

Fig. 22 illustrates the query satisfaction rate against query-based interests. Results demonstrate that INF-NDN IoT achieves a higher satisfaction rate as compared to SICN. The reason is a correlation between semantic tags and supernodes. The correlation between tags and supernodes has categorized the smart campus network into distinct contextual smart areas, each of which is identified by its corresponding tags. The contextual nature of the query-based interest name is used by the principal node in forwarding the interest to the supernode that shares the same semantic tag. As a result, semantic data retrieval is achieved which is directly proportional to the query satisfaction rate. On the contrary, SICN only supports query satisfaction rate when semantic ID is matched either with geo ID or publisher ID in the forwarding table. Otherwise, SICN does not provide semantic data retrieval if there are non-semantic IDs to match.

7) INTEREST RETRIEVAL TIME

Interest retrieval time (IRT) results of INF-NDN IoT and SICN are illustrated in Fig. 23. We increase the number of interest packets from 0 to 25 per second. Results clearly demonstrate that INF-NDN IoT shows a substantial reduction in retrieval time. The reason is that INF-NDN IoT develops query optimization mechanisms enriched with semantic tags. Optimized query directly goes to the relevant supernode with the same semantic tag and fetches precise and most relevant data. In this way, the delay associated with retrieving the identical data packet is significantly reduced compared to sending multiple interest requests. This obviates the need for multiple, separate interest requests for each individual data packet. Overall, INF-NDN IoT outperforms the SICN with 78 % to 90 % in retrieval time. On the contrary, SICN consumes additional time on every node in the calculation of matching ID. In SICN, the geographic IDs of publishers and consumers are changed with their physical mobility (not network or channel mobility).

In addition to exploring the scalability of INF-NDN IoT, we choose IRT as a scalability metric. We increased the number of interest packets in increments of 50. We observed that with each increment of 50 interest packets per second, the IRT increased by an average of 0.06 second. Therefore, it is important to note that our solution does not introduce significant overhead with the increasing number of interest packets, as measured by IRT.

The rationale for selecting IRT as a scalability metric is that the proposed scheme is specifically designed for small IoT environments. INF-NDN IoT is well-suited for networks where traffic can be categorized into distinct semantic areas, a capability that is not feasible in very large-scale networks such as [52] and [53].

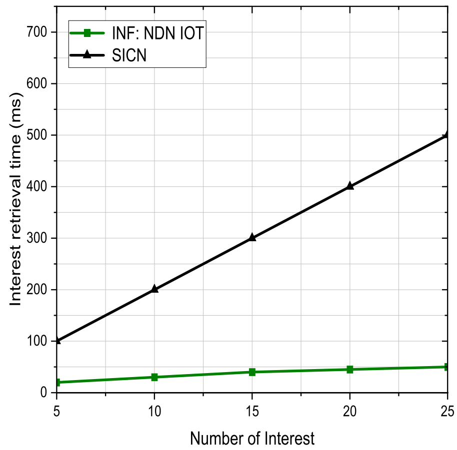


FIGURE 23. Interest retrieval time.

VII. CONCLUSION AND FUTURE WORK

In this article, we proposed INF-NDN IoT for intelligent naming and forwarding. To accomplish this, INF-NDN IoT applies NLP to optimize the content names and remove name ambiguities. Afterward, semantic information is obtained from content names, and semantic tags are assigned. In parallel, INF-NDN IoT finds the set of backbone nodes in the network called supernodes, to support semantic-assisted forwarding. These supernodes are also assigned semantic tags to develop a correlation between naming and forwarding. To demonstrate the proposed methodology's effectiveness, the principal node is responsible for the selection of supernodes and NLP-based functionalities.

As demonstrated by the results, the principal node effectively resolves any name ambiguity in the content name and forwards the interest packet based on the semantics of the content names. In addition, INF-NDN IoT reduces the average content name's length up to approximately 40.05 % and memory utilization by removing unnecessary characters from content names. INF-NDN IoT surpasses the target scheme with 70 to 90 % in retrieval time. Moreover, the proposed scheme also achieves a higher interest satisfaction rate and lower energy consumption, thus outperforming the state-of-the-art benchmark.

The architecture of INF-NDN IoT meets the essential needs for the basic requisites of NDN IoT. However, it marks just the initial stage of developing a concrete NDN-based architecture for IoT. Many important aspects remain unexplored in this scheme and will be addressed in our future work. One of these challenges is the scalability in terms of infrastructure. The scope of INF-NDN IoT is currently limited to small-scale environments, where ICN network traffic can be categorized into distinct semantic groups. This semantical categorization also presents scalability challenges in maintaining smooth and frequent communication across interdomain networks. Currently, only the principal node handles naming tasks and assigns semantic tags. Introducing hundreds of supernodes also requires hundreds of semantic tags in the network. In addition, it is not feasible to semantically categorize all nodes in a very large-scale network. Therefore, the implementation of INF-NDN IoT on a very large-scale network which includes thousands of network nodes is yet to be resolved. To address this, we plan to develop a universal semantic naming and forwarding scheme designed to facilitate effective communication across domains. We also intend to create a decentralized intelligent method in the future that can operate independently at the local level. Moreover, the semantic evaluation of CS and PIT also has been deferred to future work.

REFERENCES

[1] S. Li, L. D. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, pp. 243–259, Oct. 2015.

[2] K. Abualsaud, T. M. Elfouly, T. Khattab, E. Yaacoub, L. S. Ismail, M. H. Ahmed, and M. Guizani, "A survey on mobile crowd-sensing and its applications in the IoT era," *IEEE Access*, vol. 7, pp. 3855–3881, 2019.

[3] H. Xu, Y. He, X. Sun, J. He, and Q. Xu, "Prediction of thermal energy inside smart homes using IoT and classifier ensemble techniques," *Comput. Commun.*, vol. 151, pp. 581–589, Feb. 2020.

[4] Q. Cai, H. Wang, Z. Li, and X. Liu, "A survey on multimodal data-driven smart healthcare systems: Approaches and applications," *IEEE Access*, vol. 7, pp. 133583–133599, 2019.

[5] Z. Zhang, E. Lu, Y. Li, L. Zhang, T. Yu, D. Pesavento, J. Shi, and L. Benmohamed, "NDNoT: A framework for named data network of things," in *Proc. 5th ACM Conf. Inf.-Centric Netw.*, Sep. 2018, pp. 200–201.

[6] G. Mauri, M. Gerla, F. Bruno, M. Cesana, and G. Verticale, "Optimal content prefetching in ndn vehicle-to-infrastructure scenario," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2513–2525, Mar. 2017.

[7] A. Nurhayati, R. Mayasari, S. Ahdan, R. M. Negara, G. N. Nurkahfi, and N. R. Syambas, "Naming scheme on named data networking: A survey," in *Proc. 8th Int. Conf. Wireless Telematics (ICWT)*, Jul. 2022, pp. 1–5.

[8] Z. Sabir and A. Amine, "Connected vehicles using NDN for intelligent transportation systems," in *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, 2018, pp. 2433–2441.

[9] Z. Li, J. Liu, L. Yan, B. Zhang, P. Luo, and K. Liu, "Smart name lookup for NDN forwarding plane via neural networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 529–541, Apr. 2022.

[10] S. Shannigrahi, C. Fan, and C. Partridge, "What's in a name: Naming big science data in named data networking," in *Proc. 7th ACM Conf. Inf.-Centric Netw.*, Sep. 2020, pp. 12–23.

[11] F. A. Karim, A. H. M. Aman, R. Hassan, K. Nisar, and M. Uddin, "Named data networking: A survey on routing strategies," *IEEE Access*, vol. 10, pp. 90254–90270, 2022.

[12] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023.

[13] G. M. Raza, Z. S. Butt, S. Latif, and A. Wahid, "Sentiment analysis on COVID tweets: An experimental analysis on the impact of count vectorizer and TF-IDF on sentiment predictions using deep learning models," in *Proc. Int. Conf. Digit. Futures Transformative Technol. (ICoDT2)*, May 2021, pp. 1–6.

[14] T. Song, H. Yuan, P. Crowley, and B. Zhang, "Scalable name-based packet forwarding: From millions to billions," in *Proc. 2nd ACM Conf. Inf.-Centric Netw.*, Sep. 2015, pp. 19–28.

[15] B. S. Paris, "Time constructs: Design ideology and a future internet," *Time Soc.*, vol. 30, no. 1, pp. 126–149, Feb. 2021.

[16] A. Detti, L. Bracciale, P. Loreti, G. Rossi, and N. B. Melazzi, "A cluster-based scalable router for information centric networks," *Comput. Netw.*, vol. 142, pp. 24–32, Sep. 2018.

[17] C. Gündogan, C. Amsüss, T. C. Schmidt, and M. Wählisch, "Content object security in the Internet of Things: Challenges, prospects, and emerging solutions," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 1, pp. 538–553, Mar. 2022.

[18] A. Aboodi, T.-C. Wan, and G.-C. Sodhy, "Survey on the incorporation of NDN/CCN in IoT," *IEEE Access*, vol. 7, pp. 71827–71858, 2019.

[19] X. Tan, W. Feng, J. Lv, Y. Jin, Z. Zhao, and J. Yang, "F-NDN: An extended architecture of NDN supporting flow transmission mode," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6359–6373, Oct. 2020.

[20] R. Paul and M. P. Selvan, "A study on naming and caching in named data networking," in *Proc. 5th Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Nov. 2021, pp. 1387–1395.

[21] K. Ahed, M. Benamar, and R. El Ouazzani, "Content delivery in named data networking based Internet of Things," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1397–1402.

[22] H. Hellbrück, T. Teubler, and S. Fischer, "Name-centric service architecture for cyber-physical systems (short paper)," in *Proc. IEEE 6th Int. Conf. Service-Oriented Comput. Appl.*, Dec. 2013, pp. 77–82.

[23] C. Seales, T. Do, E. Belyi, and S. Kumar, "PHINet: A plug-n-play content-centric testbed framework for health-Internet of Things," in *Proc. IEEE Int. Conf. Mobile Services*, Jun. 2015, pp. 368–375.

[24] P. Hemanth, "Home automation using resberrypi & cloud," *Global J. Eng. Sci. Res. Manag.*, vol. 6, pp. 42–48, Jun. 2016, doi: 10.5281/zenodo.55959. [Online]. Available: https://www.gjesrm.com/Issues%20PDF/Archive-2016/June-2016/6.pdf

[25] M. I. Ayadi, A. Maizate, M. Ouzzif, and C. Mahmoudi, "Deep learning in building management systems over NDN: Use case of forwarding and HVAC control," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2019, pp. 1192–1198.

[26] D. Mars, S. Mettali Gammar, A. Lahmadi, and L. Azouz Saidane, "Using information centric networking in Internet of Things: A survey," *Wireless Pers. Commun.*, vol. 105, no. 1, pp. 87–103, Mar. 2019.

[27] T. Niu and F. Yang, "An entropy-based approach: Compressing names for NDN lookup," *IEEE Access*, vol. 9, pp. 109833–109846, 2021.

[28] A. Javaheri, S. N. S. Hashemi, and A. Bohlooli, "Hybrid naming scheme based PURSUIT architecture for smart city," in *Proc. 4th Int. Conf. Smart City, Internet Things Appl. (SCIOT)*, Sep. 2020, pp. 33–38.

[29] B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu, "M2HAV: A standardized ICN naming scheme for wireless devices in Internet of Things," in *Proc. 12th Int. Conf.*, Guilin, China. Cham, Switzerland: Springer, Jun. 2017, pp. 289–301.

[30] M. A. U. Rehman, R. Ullah, and B. S. Kim, "NINQ: Name-integrated query framework for named-data networking of things," *Sensors*, vol. 19, no. 13, p. 2906, Jun. 2019.

[31] L. Bracciale, P. Loreti, A. Detti, R. Paolillo, and N. B. Melazzi, "Lightweight named object: An ICN-based abstraction for IoT device programming and management," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5029–5039, Jun. 2019.

[32] G. Jaber, N. Pastei, F. Rahal, and A. Abboud, "Naming and routing scheme for data content objects in information-centric network," in *Proc. IEEE Open Conf. Electr., Electron. Inf. Sci. (eStream)*, Apr. 2020, pp. 1–5.

[33] T. Mochida, D. Nozaki, K. Okamoto, X. Qi, Z. Wen, T. Sato, and K. Yu, "Naming scheme using NLP machine learning method for network weather monitoring system based on ICN," in *Proc. 20th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Dec. 2017, pp. 428–434.

[34] X. Wang and S. Cai, "Edge-assisted NDN-based IoT framework with provider and consumer mobility support," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1713–1725, May 2022.

[35] F. Song, Y. Ma, Z. Yuan, I. You, G. Pau, and H. Zhang, "Exploring reliable decentralized networks with smart collaborative theory," *IEEE Commun. Mag.*, vol. 61, no. 8, pp. 44–50, Aug. 2023.

[36] T. Chatterjee, S. Ruj, and S. Das Bit, "Efficient data storage and name look-up in named data networking using connected dominating set and patricia trie," *Autom. Control Comput. Sci.*, vol. 55, no. 4, pp. 319–333, Jul. 2021.

[37] F. Karim, K. A. Abu Bakar, and N. H. Kamarudin, "Explicit routing for popular content in named data network," Social Sci. Res. Netw., Rochester, NY, USA, Tech. Rep. 4482483, 2023, doi: 10.2139/ssrn.4482483.

[38] F. Wang, H. Li, Z. Wu, P. Lu, B. Wang, Y. Yang, H. Bai, and X. Pei, "ALRP: An adaptive link state routing protocol for multi-identifier network," in *Proc. IEEE 8th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2022, pp. 677–684.

[39] M. A. U. Rehman, R. Ullah, B.-S. Kim, B. Nour, and S. Mastorakis, "CCIC-WSN: An architecture for single-channel cluster-based information-centric wireless sensor networks," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7661–7675, May 2021.

[40] A. Aboud, H. Touati, and B. Hnich, "Efficient forwarding strategy in a NDN-based Internet of Things," *Cluster Comput.*, vol. 22, no. 3, pp. 805–818, Sep. 2019.

[41] A. Djama, B. Djamaa, M. R. Senouci, and N. Khemache, "LAFS: A learning-based adaptive forwarding strategy for NDN-based IoT networks," *Ann. Telecommun.*, vol. 77, nos. 5–6, pp. 311–330, Jun. 2022.

[42] N. Akther, K. Dhar, S. M. A. Iqbal, and M. N. Huda, ''Interest forwarding strategy in named data networks (NDN) using Thompson sampling,'' *J. Netw. Comput. Appl.*, vol. 205, Sep. 2022, Art. no. 103458.

[43] E. da Silva, J. Macedo, and A. Costa, ''Context-aware routing and forwarding model for NDN-based VANET,'' in *Proc. Int. Conf. Ad Hoc Netw.* Cham, Switzerland: Springer, Nov. 2021, pp. 18–32.

[44] X. Wang and Y. Lu, ''Efficient forwarding and data acquisition in NDN-based MANET,'' *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 530–539, Feb. 2022.

[45] D. M. Blei and J. D. Lafferty, ''Topic models,'' in *Text Mining*, 1st ed., Boca Raton, FL, USA: CRC Press, 2009, p. 24.

[46] D. M. Blei, A. Y. Ng, and M. I. Jordan, ''Latent Dirichlet allocation,'' *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[47] G. Veletsianos, R. Kimmons, R. Larsen, T. A. Dousay, and P. R. Lowenthal, ''Public comment sentiment on educational videos: Understanding the effects of presenter gender, video format, threading, and moderation on Youtube TED talk comments,'' *PLoS ONE*, vol. 13, no. 6, Jun. 2018, Art. no. e0197331.

[48] J. Li, J. Wu, C. Li, W. Yang, A. K. Bashir, J. Li, and Y. D. Al-Otaibi, ''Information-centric wireless sensor networking scheme with water-depth-awareness content caching for underwater IoT,'' *IEEE Internet Things J.*, vol. 9, no. 2, pp. 858–867, Jan. 2022.

[49] Z. Kong, E. Newberry, J. Shi, T. Yu, A. Kahlert, D. Hensley. (2023). *Python-NDN*. Accessed: 2023. [Online]. Available: https://python-ndn.readthedocs.io/en/latest/

[50] NP Team, *Pyndn2*, Named Data Netw. (NDN) Project, Nat. Sci. Found. (NSF), VA, USA, 2017.

[51] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural Language Processing: Python and NLTK*. Mumbai, India: Packt, 2016.

[52] H. Touati, A. Aboud, and B. Hnich, ''Named data networking-based communication model for Internet of Things using energy aware forwarding strategy and smart sleep mode,'' *Concurrency Comput., Pract. Exper.*, vol. 34, no. 3, p. e6584, Feb. 2022.

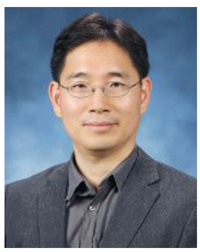
[53] F. Azzedin, ''Towards a scalable HDFS architecture,'' in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 155–161.

GHULAM MUSA RAZA received the B.S. degree in computer sciences from COMSATS University Islamabad, in 2019, major in intelligent robotics, and the M.S. degree in computer sciences from SEECS, NUST, Islamabad, in 2021, major in natural language processing (artificial intelligence). He is currently pursuing the Ph.D. degree with the Department of Communication and Software Engineering, Graduate School, Hongik University, South Korea. From 2017 to 2019, he was a Software Engineer with Snaky Solutions Pvt. Ltd. He was a Machine Learning-Based Research Assistant with the TUKL Laboratory, NUST, in 2021. He was a Lecturer with Alhamd Islamic University, Islamabad, from 2021 to 2022. His major research interests include natural language processing, the Internet of Things (IoT), information-centric networking, and named data networking.

IHSAN ULLAH received the B.S. degree in computer systems engineering from the University of Engineering and Technology, Peshawar, Pakistan, and the master's degree in computer engineering with a specialty in computer and wireless networks from the Department of Electrical and Computer Engineering, COMSATS University Islamabad, Pakistan, in 2021. Currently, he is pursuing the Ph.D. degree with the Department of Software and Communication Engineering, Hongik University, South Korea. He was a Research Assistant with the Wireless and Communication Laboratory for six months. His current research interests include NDN, underwater wireless sensor networks (UWSN), cloud computing, fog computing, vehicular networks, machine learning, and artificial intelligence.

MUHAMMAD SALAH UD DIN received the B.E. degree in information and technology from KICSIT, University of Engineering and Technology, Taxila, Pakistan, in 2013, and the M.S. degree in computer science from COMSATS University Islamabad, Pakistan, in 2016. He is currently pursuing the Ph.D. degree in computer engineering with the Broadband Convergence Networks Laboratory, Hongik University, South Korea. From 2013 to 2019, he was a Software Engineer in leading IT companies in Pakistan. His major research interests include information-centric wireless networks, vehicular networks, named data networking, wireless sensor networking, edge/fog/cloud computing, and AI for wireless communications.

MUHAMMAD ATIF UR REHMAN received the Ph.D. degree in electronics and computer engineering from Hongik University, South Korea, in 2022. He is currently an Assistant Professor with Manchester Metropolitan University, Manchester, U.K. He holds six patents. In addition to his academic background, he also has industry experience of more than six years as a Software Engineer and Architect in leading companies. His research interests include distributed networking and computing, metaverse, edge computing, the IoT, UAV, ICN/NDN, and 6G. He is a Founding Member and a part of the IEEE SIG on Metaverse. He served as the General Chair, a TPC Member, and the Session Chair for several conferences and workshops, such as IEEE ICC, iMeta, MetaNC, and ICWMC. His work has appeared in more than 50 publications in reputable venues, such as IEEE INTERNET OF THINGS JOURNAL, ACM SIGCOMM, *IEEE Intelligent Transportation Systems Magazine*, and *Future Generation Computer Systems*.



BYUNG-SEO KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. From 1997 to 1999, he was with Motorola Korea Ltd., Paju-si, South Korea, as a Computer Integrated Manufacturing (CIM) Engineer in advanced technology research and development (ATR & D). From January 2005 to August 2007, he was with Motorola Inc., Schaumburg, IL, USA, as a Senior Software Engineer in networks and enterprises. His research focuses on Motorola Inc., designing the protocol, and network architecture of wireless broadband mission-critical communications. From 2012 to 2014, he was the Chairperson of the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. His work has appeared in around 167 publications and 22 patents. His research interests include the design and development of efficient wireless/wired networks including link-adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He served as a member for the Sejong-City Construction Review Committee and the Ansan-City Design Advisory Board. He served as the General Chair for 3rd IWWCN 2017 and a TPC Member for IEEE VTC 2014-Spring, EAI FUTURE2016, and ICGHIC 2016 and 2019 Conferences. He served as a Guest Editor for Special Issues of *International Journal of Distributed Sensor Networks* (SAGE), IEEE ACCESS, and *Journal of the Institute of Electronics and Information Engineers*. He is an Associative Editor of IEEE ACCESS.