Information-Centric Networking (ICN) for Internet of Things (IoT)

A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctorate of Philosophy

Submitted By
Engr. Sobia Arshad
2014F-UET/PhD-CP-43

Supervisor
Dr. Muhammad Awais Azam

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING
UNIVERSITY OF ENGINEERING AND TECHNOLOGY
TAXILA

December 2018

Dedication

DEDICATED TO

My Parents

Mr.: M. Arshad

Mrs.: Mumtaz Begum

&

My Grand Parents

Mrs.: Sakina Bibi

Mr.: Muhammad Shafi

&

My Loving Husband

Mr.: Engr. Hammad Raza

...

Abstract

Information-Centric Networking (ICN) for Internet of Things (IoT)

Engr.Sobia Arshad
14F-UET/PhD-CP-43

Thesis Supervisor:

Dr. Muhammad Awais Azam
Assistant Professor
Computer Engineering Department

Information-Centric Networking (ICN) is being considered as a promising approach to address the issues and shortcomings of existing IP address-based networking models and cater high density of users and devices of future communication paradigm. ICN models are based on naming the contents to deal address-space scarcity and support upcoming massive connections. It allows accessing of contents vial name-based routing and caching the content at intermediate nodes to provide reliable, efficient and self-certifying contents to ensure stringent security requirements. Obvious benefits of ICN in terms of fast and efficient data delivery and improved reliability makes ICN as highly promising networking model for Internet of Things (IoTs). IoTs foremost target is to connect billions of things in a way to reduce human involvement and automate machines. Among many challenges, one key challenge is how to name and address the IoT contents and devices efficiently.

Thus, this thesis introduces the ICN (i.e., more specifically Named Data Networking (NDN) and Content Centric Networking (CCN)) for both IoT devices and contents and presents two novel naming mechanisms and one holistic forwarding scheme equipped with security, heterogeneity and scalability. Also, this thesis reclassifies IoT applications and presents their fourteen categories. From this new categorization, IoT-based Smart Campus (IoTSC) scenario is selected to design the naming and forwarding schemes due to its true representation for IoT.

Firstly, a CCN-based hybrid naming scheme is proposed which names the contents using hierarchical and flat components to support both *push* and *pull* communication and introduced two transmission modes namely (1) unicast mode and (2) broadcast mode to address loop problem associated with CCN. Simulation results demonstrate that proposed scheme significantly improves the rate of interest transmissions, number of covered hops, name aggregation, and reliability along with addressing the loop problem.

Further as an extension to the first scheme, NDN-based hybrid naming scheme is proposed which names the IoT devices and content using hierarchical, flat and attribute components to support both *push* and *pull* traffic models.

Then, on the basis of extended NDN-based hybrid naming scheme, IoT traffic types are defined using the listed activities in IoTSC. Holistic forwarding schemes are proposed for NDN-IoT consumer, producer and content routers which provide machine type communication (MTC) with push and pull communication models enabled. These forwarding schemes use another OnboardICNg security scheme which is designed to authenticate and authorize the devices to perform asked actions. These schemes enable NDN-IoT producer to send critical content or updates of subscribed content to NDN-IoT consumer through content router(s). Moreover, NDN-IoT consumer is enable to send message to perform any action or setting value of any parameter of NDN-IoT producer. These schemes are also implemented in ndnSIM and evaluated against legacy NDN in terms of interest satisfaction rate, latency and number of transmissions.

Keywords: *Internet of Things, Information-Centric Networking, Named Data Networking, Content-Centric Networking, IoT Architecture, Naming, Forwarding.*

Author's Declaration

I Sobia Arshad (2014F-UET/PhD-CP-43) hereby state that my PhD thesis titled "Information-Centric Networking (ICN) for Internet of Things (IoT)" is my own work. The work has not been submitted previously by me for taking any degree from this University "University of Engineering and Technology (UET), Taxila, Pakistan" or anywhere else in the country/world.

At any time if my statement is found to be incorrect event after my Graduation, the university has the right to withdraw my PhD Degree.

Engr. Sobia Arshad
2014F-UET/PhD-CP-43

Acknowledgements

First and foremost, I would like to express my thanks and gratitude to **Almighty Allah** who gave me strength, courage and wisdom in partial compilation of thesis. The time spent during this thesis so far was both grueling and rewarding. The rewards exceeded the pains, so I am satisfied with the overall proceedings.

Along with that, I would like to thank to my supervisor, **Dr. Muhammad Awais Azam** for his guidance, support, and patience throughout this term. Dr. Awais has been a mentor in both my academic work and life, he always sincerely guides and helps me when I am encountered with troubles and frustrations.

I would like to express my special appreciation and thanks to **Prof. Jonathan Loo** (University of West London, London, UK) for helping me to understand problem statement, algorithm designing approach to develop proposed naming and forwarding schemes. His knowledge in the field of IoT Protocol design has really helped me in understanding the problems. Moreover, his trust and high expectations pushed me not only to finish this dissertation but toward a new level of professionalism.

Also, I would like to thank **Dr. Syed Hassan Ahmed**, **Dr. Mubashir Hussain Rehmani** and **Dr. Muhammad Faran Majeed**. Their critical reviews, advice, and counsel have been of equal importance. I greatly appreciate their time and input to this dissertation.

I would also like to thank my Ph.D. research committee members, **Dr. Saleem Aslam**, **Dr. Ali Hassan** and **Dr. Syed Muhammad Anwar** for serving as my committee members and for letting my defense be an memorable moment. I want to special thank **Dr. Saleem Aslam** for your brilliant comments and suggestions, many thanks to you.

Many thanks to **University Of Engineering & Technology, Taxila, ASR/TD, CPED** for providing such a remarkable research environment for Full-time Ph.D. students. I would also like to thanks my friends (Shamila, Tehmeena, Anum and Saima) and lab mates (Aeman, Sir Sanay, Nudrat Nida, Sir Zeeshan, Fizza, Abdul Hannan) who have been with me during all the tense moments in this thesis work.

Special thanks to my loving and caring husband **Hammad Raza** for his continuous encouragement and support during this crucial phase of thesis writing. Without your unlimited sacrifices and mutual understanding, my Ph.D. journey would have never been possible.

Last but not the least my big regards and my very special gratefulness to **my parents (M. Arshad Mirza and Mumtaz Begum) and grand parents (Sakina Bibi and Muhammad Shafi)** for their love, support, and sacrifices. Their continued encouragement throughout the term is what made it possible for me to complete this work. My love and gratitude is extended to my brother **Ali Haider** and sisters (Nadia, Bushra, Mariyam, Muqaddas, Amal and Eman) for their love and support.

Publications

1. **Arshad, S.**, Azam. M.A., Ahmed. S.H., and Loo. J., Towards Information-Centric Networking (ICN) Naming for Internet of Things (IoT): The Case of Smart Campus, In Proceedings of ICFNDS 17, Cambridge, United Kingdom, July 19-20, 2017, 6 pages.
2. **Arshad, S.**, Shahzaad. B., Azam. M.A., Loo. J., Ahmed. S.H., and Aslam. S., Hierarchical and Flat-Based Hybrid Naming Scheme in Content-Centric Networks of Things, Published in, IEEE INTERNET OF THINGS JOURNAL, 2018, 11 pages.
3. **Arshad, S.**, Azam. M.A., Rehmani. M.H., and Loo. J., Recent Advances in Information-Centric Networking based Internet of Things, Published in, IEEE INTERNET OF THINGS JOURNAL, 2018, 31 pages.
4. Hannan, A., **Arshad, S.**, Azam. M.A., Loo. J., Ahmed. S.H., Shah. S.C., and Majeed. M.F., Disaster Management System Aided by Named Data Network of Things: Architecture, Design, and Analysis, Published in, Sensors (Basel, Switzerland), 2018, 20 pages.
5. **Arshad, S.**, Azam. M.A., Loo. J., and Majeed. M.F., Holistic Forwarding for Information-Centric Network of Things-based Smart Campus, Submitted in, IEEE COMMUNICATION LETTERS, 2018, 4 pages.
6. **Arshad, S.**, Azam. M.A., Loo. J., and Majeed. M.F., Named Data Networking Holistic Naming and Forwarding for the Internet of Smart Campus Things, Submitted in, IEEE INTERNET OF THINGS JOURNAL, 2018, 9 pages.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

IoTs aim to connect each and every device with the Internet, so that these devices can be accessed at any time, at any place and by any path (i.e., from any network) [1]. IoTs canopies enchanted objects like smart washing machines, smart refrigerators, smart microwave ovens, smartphones, smart meters and smart vehicles. Connectivity of these smart objects with the Internet enables many valuable and remarkable applications like smart home, smart building, smart transport, digital health, smart grid and smart cities. When billions devices connects to the Internet, generation of large amount of data is an apparent consequence. Besides that, this data combines with the data that produces through, for instance Facebook likes and Youtube videos etc. Thus efficient data discovery and access put more constraints on the underlying TCP/IP architecture and eventually raises many issues. Among these, from device perspective, one is naming and addressing every device [2]-[3]. As IPv4 addressing space is exhausted, IPv6 address space may also exhaust in future. In addition, IPv6 address is long and it is less suitable for communication through processing power constraint-oriented devices like wireless sensors [4]-[5]-[6]. Therefore, efficient naming and addressing schemes for billions of devices (and contents) are not ideally available in IP-architecture. On the other hand, every device has different specifications and constraints raising another issue of heterogeneity. IoTs comprises on heterogeneous devices and most of the devices are tiny, low power, limited memory, low cost and constraint-oriented wireless sensors. These devices are usually known as smart devices. Due to low memory and low battery life, data can become unavailable most

of the time. Thus solutions like in-network caching are missing in naive IP based networking. IoTs applications like smart home, smart town, smart grid and smart health requires more security and extra privacy in terms of data accessed by these devices and their usage [7]. Moreover, some IoTs applications, for instance, VANETs, MANETs and smart transport requires better mobility handling [8]-[9]. From data perspective, mostly IoTs application users are more interested in getting the updated information rather than knowing the address of information source. For instance IoT devices, especially in the domain called wireless sensor networks (WSN), have specific purpose to harvest information and at the large scale [10]. Every device has to perform some specific task, for example temperature sensors measure temperature from their surroundings and does not perform word processing task that a general purpose computer does. Any user of temperature measurement application is interested in current temperature of a certain area instead of the temperature value of a specific sensor. To fulfill these above mentioned requirements and due to recent trends about IoT architecture have prompted many research organizations to initiate multiple projects. Therefore many evolutionary (or dirty slate) approaches are being explored for IoTs, for instance IPv6 based 6LoWPANs [11]-[12]-[13].

1.1.1 Clean-Slate VS. Dirty Slate Architectures

Among these, most of the projects are working under Internet Engineering Task Force (IETF). IETF projects are designing protocols for constraint-oriented devices based networks. The Constrained RESTful Environments (CoRE) [14] group designed a framework for smart applications to work efficiently on IPv6 based constraint-oriented smart devices. Constrained Application Protocol (CoAP) [15] is a major achievement that accomplished under CoRE working group. CoAP is a lighter version of HTTP protocol. It is mainly designed for low power devices forming constrained networks. CoAP also supports various caching forms that was mentioned in REpresentational State Transfer (REST) protocol. CoAP runs over UDP to provide better communication among resource-oriented devices. IPv6 over Low Power Wireless Personal Area Networks working group (6LoWPAN-WG) [16] has focused on 6LoWPANs. This group works for adaption of IPv6 over IEEE 802.15.4 based networks. 6LoWPAN

group also works for IPv6 header compression to efficiently run over low power devices. Routing Over Low power and Lossy networks working group (ROLL) [17] mainly focuses on developing routing strategies and self-configurable mechanism in low power networks. Low power and Lossy networks (LLN) made up of many embedded devices which include limited power and memory devices. LLN provides an end to end IP based solution for routing over these networks. 6LoWPAN-WG will work closely to ROLL. Sometimes situations can happen in IoT when constraint-oriented devices are required to communicate with each other without any gateway. Therefore, IETF has designed IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [18] for communication between constraint-oriented devices. RPL provides support for point-to-point and multipoint-to-point and point-to-multipoint traffic patterns. The Light-Weight Implementation Guidance (LWIG) working group [19] is focusing to build minimal and inter operable IP protocol stack for constraint-oriented IoT devices. And the Thing-2-Thing Research Group (T2TRG) [20] aimed to explore the factors that will influence the process of turning IoT into reality. T2TRG will investigate and list the issues to form the internet through which low power constraint-oriented devices can communicate to each other using M2M communication style and with the global Internet. Moreover, the European Telecommunications Standards Institute (ETSI) [21] is working on the standardization of data security, management, processing and transport for IoT on the basis of IPv6. However, more details about IoT projects and protocols can be found on [22].

1.1.2 Motivation

Above mentioned projects for IoT architecture lies under all-IP architectures' umbrella. And IP-based networking is inherently designed for host-to-host communication where location (e.g., address) of host plays a vital role, but this location-dependent design creates certain bottlenecks such as efficient information retrieval and delivery. Also, IP networking requires additional protocols to support privacy and security of sensitive data, scalability, mobility and heterogeneity of nodes. Consequently, traditional IP-based networking is less suitable for these IoT devices and applications. Hence, to provide efficient connectivity among low power IoT devices, a novel networking model like ICN, holds much promise [23]. Due to this, IETF has also

started ICN research group that will help to evolve IP-based architecture [24]. ICN is a promising candidate for the future Internet foundation. ICN primary characteristics include in-network caching, naming the contents, better mobility, improved security and scalable information delivery which are naturally suitable for IoT applications. So far, there are nine major architectures proposed under the concept of ICN including DONA, CCN [25], PURSUIT [26], NetInf [27], CURLING [28], CONET [29], MobilityFirst [30], C-DAX [31] and Green ICN [32]. Among these ICN based architectures DONA, SAIL, COMET and CONVERGENCE, CCN all are dirty-slate while MF, PURSUIT and NDN are clean-slate architectures. CCN (NDN) is prevailing approach among other ICN based proposed architectures [33]. ICN based hourglass architecture provides us thin-waist like TCP/IP [34]. ICN mask over TCP/IP network layer or MAC layer is narrow enough to accommodate more devices or networks. Current literature [35]-[36] argue that ICN seems to replace IP, rather we believe and foresee ICN as a model which can also act as an overlay network sitting on IP network. In fact, CCN is a layer that masks the need of associating content with the IP address instead by name. The actual content delivery still requires TCP/IP interface or direct MAC (layer 2) interface. CCN could also be applied just above MAC layer especially in WSN. ICN's striking feature in-network caching can handle efficiently the issue of information delivery from dead (unavailable) device due to low battery life by caching contents at intermediate nodes. Also it can minimize retrieval delay even in case of alive devices through the use of caching. While naming the contents can resolve the address space scarcity issue of IPV4 and can enable scalability in an efficient way. It also offers better name management and easy information retrieval of huge data produced by IoT applications. Moreover, mobility handling provides better hand-off for mobile devices like mobile phones and vehicles. ICN's self-certifying contents provide more security to data rather than securing the hosts [37]-[35]. That's why in this research we survey and evaluate ICN for IoTs.

1.2 Problem Statement

Internet of Things (IoTs) aims to provide global access to information and global connectivity among smart constraint-oriented devices. IoT canopies enchanted objects

like smart washing machines, smart refrigerators, smart microwave ovens, smartphones, smart homes, smart vehicles, smart towns and smart cities. IoT aims to connect each and everything (i.e., device) with the internet so that these devices can be accessed at any time, at any place and by any path (i.e., from any network). This trend of connecting every device to the internet will eventually raise the issue of naming each and every device. Therefore IPv4, as well as IPv6 addressing space can not be enough for the case of IoTs. Moreover, IoT devices can be tiny, low-power, low-cost and constraint-oriented devices. These devices are usually known as smart devices. Smart devices have low processing power and small memory. Therefore, these devices cannot support traditional heavy OSI model of networking. Additionally, every device has to perform some specific task like temperature sensor measures temperature and does not need to perform tasks like word processing which a general purpose computer does. Consequently again, traditional IP-based networking is not suitable in this condition of IoT. To support these, IoTs needs an efficient unified network architecture which provide addressing and naming of devices and contents. In last 3-5 years many approaches (which we discuss above as clean-slate and dirty-slate) are considered to solve the issue of naming and addressing. Moreover ICN models provides hierarchical, flat-based naming schemes for content naming. But for ICN-IoTs, there is not any single scheme proposed which fulfills above-mentioned requirements of IoTs. To solve the issue, we intend to develop ICN (NDN)-based architecture for IoT. This ICN (NDN)-based IoT architecture is capable of naming the IoT contents and devices. Moreover, ICN-IoT architecture is also enabled to forward content from IoT producer, consumer and content router in an optimal way. Proposed naming and forwarding schemes provides security, scalability, heterogeneity and interoperability and machine type communication (MTC).

1.3 Aim and Objectives

1. To read and develop supporting theory of ICN implementation in IoT.

2. To evaluate the influence of ICN characteristics, network architecture and infrastructure on IoT.

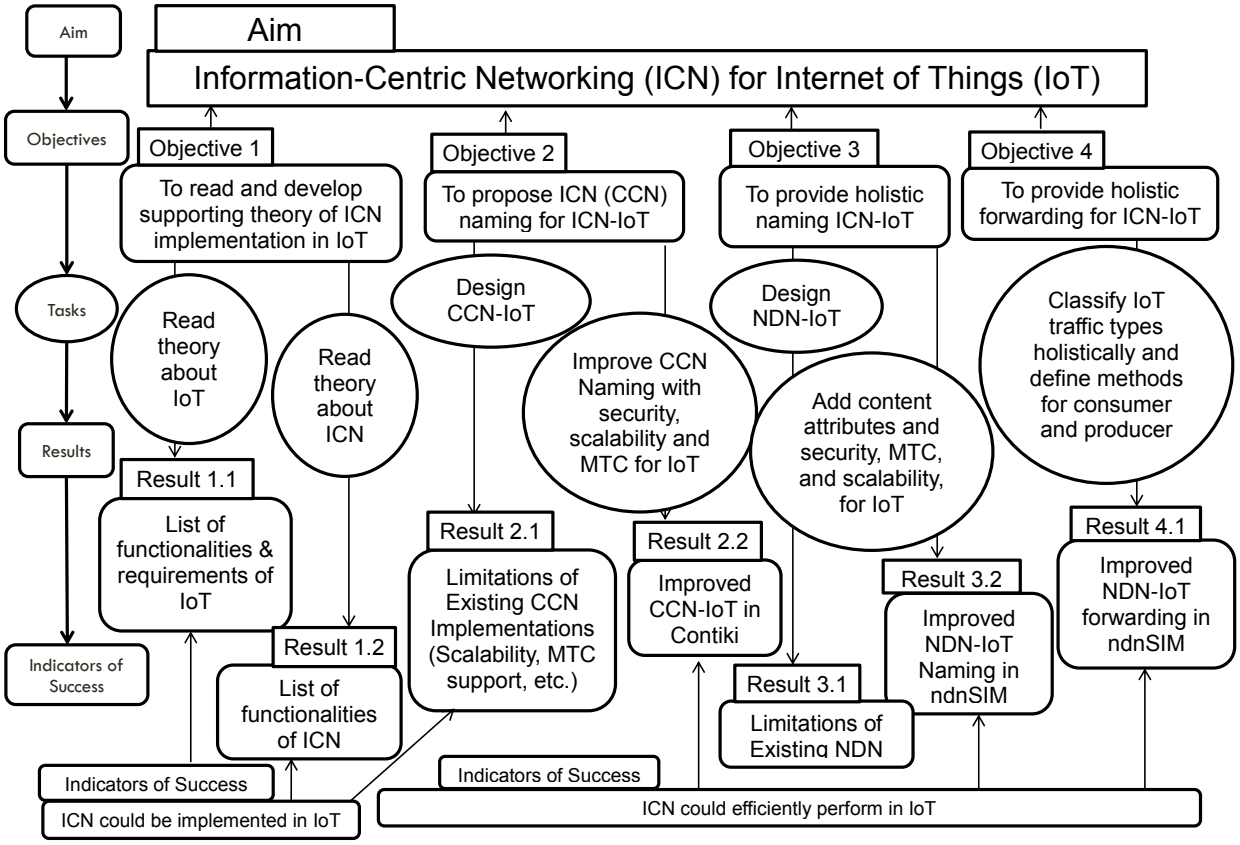

Figure 1.1: Aim and Objectives.

3. To propose efficient ICN (CCN, NDN)-based naming and forwarding solutions for IoT.

4. To evaluate the QoS performance of the proposed solutions for IoT network against legacy NDN.

1.4 Research Contributions

The major aim of this research is to estimate ICN based architecture for IoT. This major goal of research is achieved through following research contributions:

1. This thesis identities IoT requirements and marks/maps against ICN characteristics. Among ICN models, we identify CCN (NDN) model is more suitable

for IoTs. Moreover, ICN (NDN)-based three layer architecture is modified for IoTs.

2. This thesis provides details of proposed CCN-based hierarchical and flat based hybrid content naming (CCN-HNS) schemes which support both push and pull communication styles along with providing improved security, scalability, throughput and delay. Proposed schemes are implemented in famous IoT OS Contiki with Cooja simulator and in known NDN simulator ndnSIM.

3. NDN-based hybrid content naming scheme (NDN-HNS) is proposed which incorporates hierarchical, flat and attribute-based components and support IoTSC MTC traffic types along with improved interest satisfaction rate, delay and number of transmissions. NDN-HNS is utilized to propose basic forwarding algorithms (NDN-HNFS) for consumer, producer and content router. IoTSC is simulated and implemented NDN-HNS in ndnSIM. NDN-HNS is the extension of our scheme CCN-HNS.

4. NDN-HNS is utilized to propose holistic forwarding algorithms (NDN-IoT-SMTC) for consumer, producer and content router which support MTC, scalability, secure forwarding through OnboardICNg (i.e., authentication and authorization scheme). NDN-IoT-SMTC is simulated and evaluated in ndnSIM against legacy NDN in terms of interest satisfaction rate, delay and number of transmissions, and found former the optimal one.

1.5 Dissertation Organization

We have consolidated thesis related stuff in the following way;

1. **Chapter 2** describes briefly IoT, ICN, and ICN for IoT and discusses ICN naming for IoT in detail. Moreover, research problem formulation is also presented in this chapter. This is our **first** contribution.

2. **Chapter 3** presents research methodology and ICN-IoT architecture along with abstract level description of proposed schemes and contributions.

3. **Chapter 4** discusses our **second** contribution which includes the details of CCN-based hybrid naming scheme, its simulation, performance evaluation and comparison results with the state-of-the-art schemes. The work presented in this chapter, is implemented for IoT Smart Campus (IoTSC) in Coooja Simulator.

4. **Chapter 5** covers our **third** contribution which discusses the details of NDN-based hybrid naming scheme (NDN-HNS). Updated classification of IoT applications and description of NDN-HNS components are presented. Then proposed algorithms for message type determination and basic forwarding algorithms (NDN-HNFS) are discussed along with simulation scenarios and performance evaluation of NDN-HNFS for both PLTC and PHTC traffic models against legacy NDN model.

5. **Chapter 6** describes our **fourth** contribution which is holistic forwarding algorithms NDN-IoT-SMTC for consumer, producer and content routers. NDN-HNS based IoT holistic traffic models (PLTC and PHTC) and activities list for IoTSC are described. It also describes NDN-HNS based holistic forwarding algorithms NDN-IoT-SMTC for IoTSC, simulation scenarios for PHTC and PLTC (which we have implemented in ndnSIM) and performance evaluation.

6. **Chapter 7** concludes the dissertation and provides future research directions.

1.6 Chapter Summary

In this chapter, firstly, a brief introduction to thesis topic background and motivation is presented. Then, research challenges along with aim & objectives are described. Then, we summarize the major contributions we made in this work. Finally, organization of this report is described.

Chapter 2

Literature Review

This chapter is presented with fourfold purpose. Firstly, we present and discuss IoTs. Secondly, we discuss ICN w.r.t IoTs. Then thirdly, we present general ICN-based IoT related research efforts. Finally, we present naming literature survey following the summary of chapter.

2.1 Internet of Things (IoTs): An Overview

This section is presented with twofold purpose. Firstly, we present an overview of IoTs including background, basics and components involved in IoTs life cycle. Secondly, we list and describe resulting IoTs architecture requirements. However, our aim is not to survey and discuss IoTs in depth rather we illustrate it to highlight issues and identify architecture requirements. These IoTs requirements mapped against ICN striking features to show ICN significance for IoTs in the following section.

2.1.1 Brief Background

With the miniaturization and rapid production of smart devices in different domains like RFID and WSN caused their huge and frequent usage. Data that these smart devices collect is being shared and transmitted at an amplifying rate with the help of Internet enabled smart devices.

Eventually this trend of connectivity among smart devices with the Internet had born the concept of Internet of Things (IoTs).

2.1.2 IoTs Definitions

In 2010, authors of [2] defined IoTs as multi-vision paradigm that cover things oriented, Internet oriented and semantic oriented visions. This semantic oriented vision is the important one that exhibits the real power of IoTs. Moreover they state IoTs as a disruptive technology.

IERC [1] defined broadly IoT as global and dynamic network that can configure itself by employing interoperable communication protocols and things include both virtual and physical things that carry unique addresses along with physical and virtual features.

According to [3] IoT and smart environment involves devices that are referred by sensors and actuators. Through this perspective innovative applications can be developed that involves definitely information representation, data analytics, ubiquitous sensing with the help of cloud computing to form a unified network [3]-[38].

Thus IoT is introduced as a novel concept that is being supported by existing technologies. IoTs is manifold technology combining transport, health, building, industrial automation, environment, agriculture, RFID and personal sensors. Moreover IoT mainly consists of conversion mechanisms for data collected from smart devices into information though web of things to provide knowledge to us. And IoTs evolutionary process of correlative technologies involves: machine type communication (MTC) , WSNs, sensor web enablement, sensor web, web of things and semantic sensor networks [38].

2.1.3 IoTs Connectivity Types

As IoTs is the connectivity of things through the unified Internet. Things can be humans and smart machines of any sort. These things can connect in three ways: i) Machine-type-Communication (MTC), ii) Machine-to-Human (M2H) and iii) Human-to-Human (H2H). Moreover this is illustrated in the upper portion of Fig. 2.1.

- Machine-type-Communication (MTC) connectivity: This sort of connection occurs when a machine (object) wants to share its collected data to another machine or when a machine controls another machine. For example washing machine controller asks its buzzer to turn on or off when timer triggers. This MTC connectivity is the soul of IoT environment. MTC was formerly known as M2M

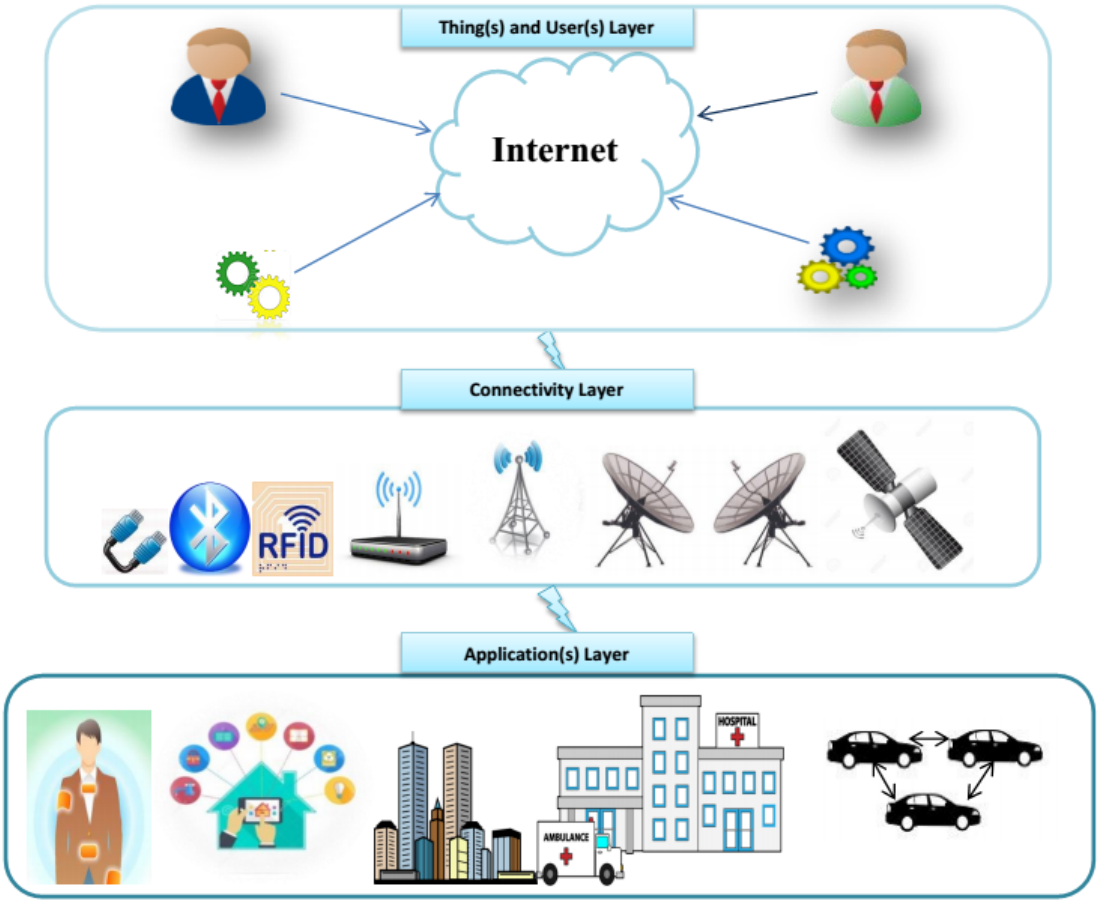

Figure 2.1: Internet of Things (IoTs): Connectivity Types, Internet Technologies and IoTs Smart Applications.

connectivity. It fulfills main purpose of IoTs to automate the life of an individual to make it better. This broader connectivity can be further elaborated in following categories that actually involves this MTC connectivity.

- Machine-to-Human (M2H) connectivity: It happens either a machine has to deliver some important information or response of any query towards human or when human needs to connect with machine to control it. For instance when house owner enters the house, garage door confirms his/her entry by sending a message to the owner.

- Human-to-Human (H2H) connectivity: This way of connection is already in use where a human is connecting with other human to build and enjoy social relations. They share photos and every moment of their life with other humans.

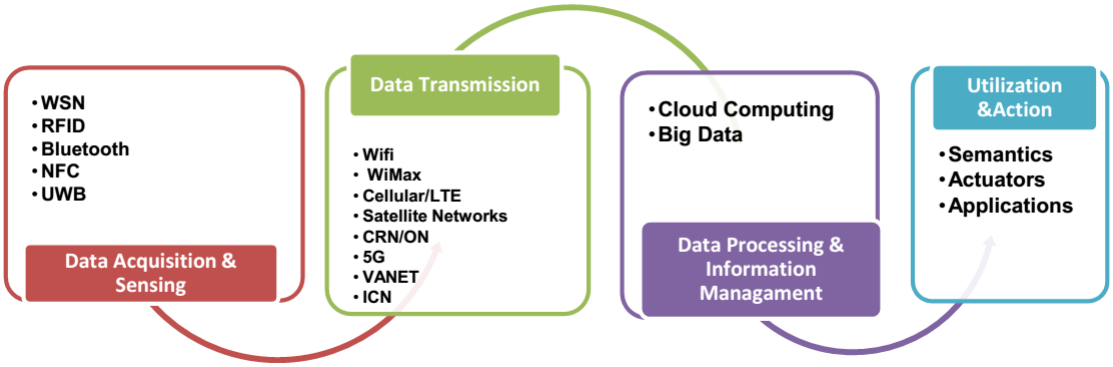

Figure 2.2: Phases in IoT and Corresponding Enabling Technologies

For this sake they need to connect with the Internet.

2.1.4 IoTs Life Cycle and Corresponding Elements

Resulting IoTs life cycle and corresponding enabling technologies are shown in Fig. 2.2. IoTs life cycle involves usually four phases: i) Data acquisition and sensing, ii) Data transmission, iii) Data processing and information management and iv) Utilization in user applications and actions. These phases are briefly described below in following sub-sections.

These major IoT working phases and corresponding elements related literature is listed in Table. 2.1.

2.1.5 IoTs Architecture Requirements

Specific requirements and challenges [2, 10, 5] introduced by IoT network architecture outlined and given below:

2.1.5.1 Scalability

As IoTs envisions not only connecting networks and corresponding devices but enabling low power devices in billions to connect through Internet. Thus, it imposes new challenges over underlying architecture in terms of scalability. IoTs architecture needs to support billions devices in efficient way. Current solutions like IPV6 has huge address space that can serve IoT devices. Although in future, addressing the IoT devices is not the only issue. Another case is large amount of data that is being

Table 2.1: IoTs Phases and Corresponding Technologies

IoT Phase		Components and Reference(s)
Acquisition and Sensing		RFID[39, 40, 41, 42] WSN [43, 44, 45, 46] Bluetooth[47] NFC[48] UWB[49, 50, 51]
Data Transmission	Current Enabling Technologies	Bluetooth[47], Ethernet[52] Wi-Fi[53, 54, 55, 56] Wi-MAX MANETs[57, 58, 59, 60] Cellular Networks[61, 62, 63, 64, 65, 66, 67] Satellite Networks[68, 69, 70]
	Future Enabling (or Enabled by IoTs) Technologies	CRN[71, 72, 73, 74, 75, 76] VANETs[77, 78, 79, 80, 81, 82] 5G[83, 84, 85, 86, 87] ON[88, 89] PLC[90, 91, 92, 93]
Data Processing and Info. Management		Cloud Computing[94, 95] Big Data[96]
Action and Utilization		Semantics[97, 98, 99, 100, 101, 102, 10] Actuators[10] Applications[1, 10]

produced by IoT devices needs better and efficient scalability management. Therefore, there is need to explore IoTs network architecture in terms of scalability and it should be scalable to content access and network efficiency.

2.1.5.2 Mobility

Number of mobile devices connecting to the Internet exceeds the stationary nodes. Mobile devices like tablets, smart-phones have small screen and limited battery life. Some IoTs applications involves and requires anytime, anywhere connectivity, in which users want to check their emails and/or make calls at anywhere, anytime. To provide fast, reliable connectivity and make data available at everywhere, network architecture should support seamless mobility and roaming.

2.1.5.3 Security and Privacy

As in some IoT scenarios like smart health and smart hospital; data that needs to be transmitted, is highly sensitive. If any hacker tries to change it, it can lead to alarming condition. To enable IoT efficiently, it should provide authorization, confidentiality and integrity. Standards are needed to specify the data access policies like who can access the data and who cannot. Take the example of smart home [103] where the detail of pizza ordered by house owner is required by pizza shop to charge the payment. If this detail is shared to his doctor or insurance company, this can effect user privacy. As insurance company is not the tentative user and could use the private data in wrong way. However, privacy must be ensured via some access policies.

2.1.5.4 Naming and Addressing

IoT consists of billions of tiny, low-power, constraint-oriented devices which needs unique names or addresses to get recognition in the network. If we talk about a single nano-network which may contain thousands of nano-nodes and then interconnection of many nano-networks would require complex IDs or addresses. Although large address space is available in IPv6, it may help addressing and naming problem of IoT devices. But for constraint oriented simple devices it would be complex to process long address for a very small communications thus resulting the wastage of resources. Further IoTs contents being produced and processed at very fast speed. In addition these there can be many versions or values against any single content with different time stamps. Naming these rapidly produced contents is issue for IoTs. Thus still a larger and permanent naming scheme and addressing space is highly needed for IoTs contents.

2.1.5.5 Heterogeneity and Interoperability

As we have seen above that RFID tags and smart sensors mainly build IoTs. Further smart sensors being major components of IoTs offer applications in many-sides. These devices are heterogeneous in nature and usually varies in specifications like in memory size, processing power and battery life. Moreover communication between these sensors is carried out by different underlying technologies (wired, wireless, cellular,

bluetooth, 4G, LTE, CRN, opportunistic networks). Thus heterogeneous technologies are involved in communication. Therefore network architecture is required to support heterogeneity among device specifications and different underlying communication technologies and techniques in an interoperable way.

2.1.5.6 Data Availability

In the current TCP/IP-based architecture, whenever a node moves from one location to another, data that it assumed to provide becomes unavailable. Same case also occurs when some device runs out of battery and is not capable to forward data. In addition, Internet users cannot receive data at time due to occurrence of denial of service (DoS) attack. DoS occurs because the current Internet architecture cannot look or inspect data according to request during data transmission. Consequently, methods like in-network caching are required to make data available with absolute certainty.

2.1.5.7 Energy Efficiency

As obviously billions devices need huge amount of energy to build IoTs applications. Moreover, most of the smart devices are low in battery life such as wireless sensors. Thus energy efficient mechanisms are required to make this universal connectivity possible in the form of IoTs.

2.2 Information-Centric Networking (ICN) Overview w.r.t IoTs

This section fulfills three purposes: Firstly, we provide a brief introduction of ICN concept along with its suitability for IoT. Secondly, we discuss ICN supporting IoT features and provide their mapping against IoT architecture requirements, and lastly we describe briefly ICN-based proposed architectures w.r.t their naming, caching, security and mobility feasibility for IoTs.

2.2.1 Limitations of TCP/IP Model and Importance of ICN for IoTs

As TCP/IP was traditionally designed to connect limited number of computers, to share limited and expensive network resources with limited address space at network layer. And inherently it is not designed to fulfill IoTs requirements efficiently. Moreover, IoTs huge data put additional requirements on the underlying architecture like data dissemination, security, mobility and scalability. In addition, flash crowds are the obvious consequence of todays Internet usage [34, 35, 104, 105, 106]. Flash crowd is a situation which occurs in the Internet when large number of Internet users request for a particular information item. As a consequence, flash crowds increase network traffic for any particular server (i.e., originating and providing that specific information item) [107]. To minimize flash crowd, ICN provide and support a much-needed characteristic named: *in-network caching* which minimizes traffic load on original data producing server while caching the data on intermediate routers. In this way, intermediate routers can send required data on behalf of original producer and as a result, load can be minimized. From both, today's Internet and IoT context, all users just need data even without knowing the producer of that data. More specifically, in IoTs, (i.e., where any specific node can act as producer and consumer at the same time) for example; when an accident occurs somewhere on any road, that vehicle want to inform incoming vehicles about this incident. As a result, flash crowd occurs because only one vehicle is providing the data about that incident. Data can become unavailable due to end of batteries of many sensors located in that producer vehicle. But with the help of ICN in-network caching, any vehicle can provide data who cached that information item while reducing so-called flash crowd situation.

Moreover, in native ICN, information (i.e., content) is named independent from its location so that it can be located anywhere globally. Naming the data and devices makes ICN more suitable for IoT as it can combine billion of devices and huge information contents. As IoT receiver of information is more interested in data rather than its location. ICN supports receiver driven communication making the communication under full control of receiver. Push type communication can be provided using beacon messages [108]. Furthermore data can only be accessed whenever receiver explicitly requests a data. ICN offers in-network caching making it ideal for low power devices.

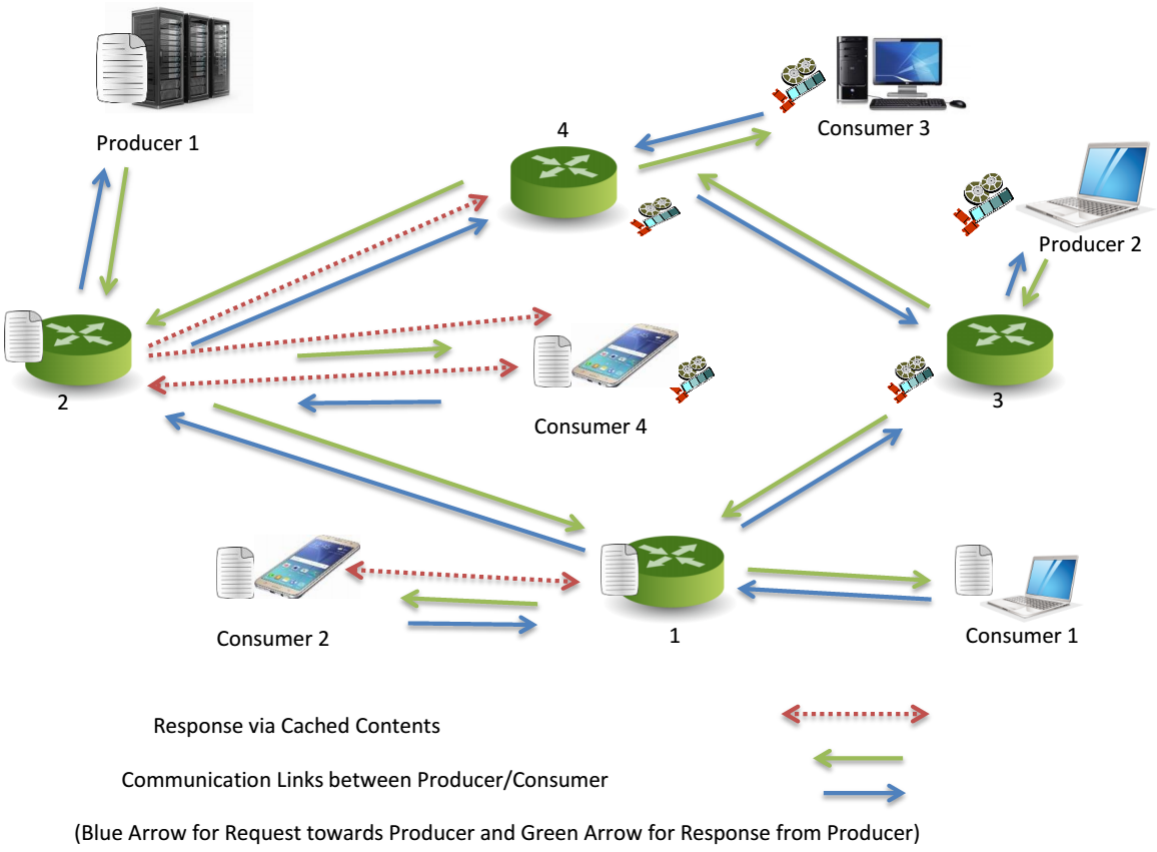

Figure 2.3: ICN Operation: Consumer Requests for a Specific Content by Nearest Routers (1,2,3,4) and Producer Replies and Intermediate Nodes which Cache the Content can Fulfills Further Requests Through Cached Contents Rather Than Sending Request Towards the Original Producer

Data is found on the basis of its location-independent name. This provides opaque communication between sender and receiver making it more secure.

Due to ICN's fruitful features and to solve IoT global connectivity efficiently, research community has started to propose different solutions to overcome issues presented by TCP/IP. As a result of these efforts, ICN is appeared as promising approach for IoTs [23]-[36]-[37]-[109]-[110]. Details of ICN (specifically NDN) operation is shown in Fig. 2.3.

2.2.2 ICN Features Reevaluated for its Suitability for IoTs

In this subsection, we discuss and elaborate main features of ICN proposals to evaluate their usefulness for IoTs.

2.2.2.1 Named Data

Naming the data is the most powerful feature of ICN proposals. As Internet users are more interested in information rather than its location. To fulfill this, ICN proposals are designed to provide named data independent of its server name. Moreover, TCP/IP is a host centric network model that was basically designed to share expensive resources. Instead today's Internet usage is not only limited to share network resources but for information dissemination with fast delivery rate to billions of devices and these are the major requirements that Internet should support. ICN resolves these above mentioned issues by just naming the data [35]-[104].

Naming the data makes, in-network caching easy and efficient as named data is easy to manage and search. Therefore, naming reduces information retrieval delay. In addition, access strategies works efficiently if these know what (named) data is being transmitted to which node. Moreover, multi-cast forwarding outperforms because of aggregation of name base requests [36]-[111].

Naming defines the structures of how to name a data. Currently two naming schemes are there that ICN models support: hierarchal and flat. Names can be human readable or not, but they are assumed to be location independent [35].

Naming (the data) supports IoT applications as IoT users are interested in information. For instance when we query about traffic condition at a specific road. Any vehicle (node) who has data, can response to our query.

2.2.2.2 Receiver Driven Communication

This is unique and a very beneficial feature of ICN that communication initiates only when a consumer needs. This feature makes ICN different from the current network architecture in the way that it is completely receiver driven. It is the responsibility of network to search for the requested data. To enquire a data, receiver places a request excluding its location or server name. Then network finds the best possible source for the requested information.

Two approaches are there in ICN, through which receiver can receive information named: coupled and decoupled approach. In coupled approach, name resolution and routing are aggregated and data is routed by following reverse path towards consumer. In decoupled approach where name resolution and routing are performed independently, any path other than, that request message follows from consumer (requesting node) to information provider, is utilized to forward response towards consumer.

As IoT devices and networks are resource constraint, receiver driven communication helps to save the network resources in the way that it can reduce number of transmissions. This is because, data will be only transferred when IoT user subscribes to an IoT device. For example, in case of smart campus, students get information about class timings from smart notice board. Administration authorities update time table on a central platform (smart notice board). Students who have subscribed to notice board can get time table updates. Moreover mobile nodes when change location, can subscribe to new nearest routers. Due to receiver driven communication, these devices can resume their ongoing transmissions.

Although sender driven communication is inherently not supported by ICN. Thus besides the advantages of receiver-driven communication, sender driven communication can be provided through the methods presented in [108]. Mostly sender driven communication is required only when some incident occurs. Therefore, this required type of communication can be provided through beacon messages [108].

2.2.2.3 In-network Caching

To reduce workload on the nodes, ICN supports in-network caching. In network caching helps in case of flash crowds when large number of nodes request for same data at the same instant (as discussed earlier in this Section). In situations like flash crowds, ICN caching outperforms the current network. When a request is fulfilled by any router, its corresponding data is cached on all intermediate nodes. Intermediate nodes can now response to that query while acting as server. Two types of in-network caching mechanisms are there:

1. On-path: caching is done on the path taken by request message.
2. Off-path: caching is performed out of the path.

In coupled approach, off path caching is supported by routing system. While in decoupled ICN architecture, name resolution system is responsible to provide off path caching. Caching always provide fast information delivery.

Moreover, caching enabled ICN-based IoTs applications improves network performance in terms of fast delivery and reduces traffic. Incoming requests could be respond through cached data in case of network failure and node unavailability (IoT constaint-oriented nodes runs out of battery frequently) [35].

2.2.2.4 Mobility

Now wireless mobile devices are more in number than static devices, that do not change their location. Our daily life gadgets like mobile phones, tablets and automated cars are wireless and mobile. Basically host-centric communication model TCP/IP is not meant to support mobile devices. Mobile devices need to change their IP addresses. Solutions like mobile-IP patch over TCP/IP increases complexity to deal with mobile devices and presents 'three tunnel communication' problem [35].

In ICN mobility is efficiently provided by employing publish subscribe model. Any mobile node subscribes the network for the information it needs. Publisher advertises the availability of information to the network. Other nodes present in the network that match the subscriptions with the publications helps to direct communication between publisher and subscriber. Moreover, it is possible that either publishers are unaware of their subscribers or subscribers do not keep record of their publishers.

A mobile device subscribes to the network and receives data. When mobile device's handoff occurs, network shifts device connectivity to a cached node rather than the original producer. Although producer mobility is not addressed yet as it is more complex. But through in-network caching, information may be provided through cached data on intermediate nodes in the situation when information provider changes its location. These all features of ICN makes it best communication model for mobile (consumer) devices.

Therefore efficient ICN mobility support can play an important part for IoT applications like smart transport and smart vehicles.

2.2.2.5 Security

As Internet was deigned to connect computers and forward any data over the network. This feature allowed hackers, spammer and attackers with bad intentions to transmit their own information while jamming the network traffic for rest of the users causing DoS. There were many solutions proposed and used to cater this security and privacy issues like IPSec. protocol. But they do not perform efficiently in scanning network deeply to find such malicious activities and to stop these activities. Major issue arises from application layer and network layer separation and disconnection that provide opaque packets difficult to inspect. On the other hand DPI (Deep Packet Inspection) is expensive solution to inspect individual packets. Thus TCP/IP secures terminals but not data.

On the other hand, in ICN, data packets are transparent and easy to check whether these packets are valid or from some spammer. Thus ICN data naming lessens DoS attacks and vague transmissions. Moreover no data can be transmitted unless a receiver requests for it. Inspection of named request message makes it easy to identify that response being transmitted is valid or not. ICN flat names may not human readable and short but provide self-certification. While ICN hierarchical names are human readable, long. Therefore long names need a trust manager to verify whether data being transmitted is according to interest message or not [35].

IoT applications like smart home, and e-health require high security of transmitted data. To provide this security in better way, ICN self-certified names can play a vital role.

2.2.3 IoT Requirements Mapping to ICN Characteristics

IoT applications that requires scalability in terms to support billions of IoT devices and huge quantity of contents can be build using ICN characteristics like naming the contents, in-network caching and content-based security. ICN naming and name resolution can be efficiently used to provide billion of addresses and names to IoT devices and contents respectively. To support IoT applications involving mobile devices, ICN receiver-driven communication feature along with flexible naming the contents and location independence can play an important role to make hand-off easy for mobile devices. Moreover, ICN in decoupled mode can perform easy re-registration after a

Table 2.2: IoT Requirements Mapping to Supporting ICN Features

Sr#	IoT Requirement(s)	ICN Supporting Features
1.	Scalability	Naming, In-Network Caching, Content-based Security
2.	Naming and Addressing	Naming and Name Resolution (Coupled and Decoupled mode)
3.	Mobility	Decoupled Mode, Naming, Receiver Driven, Location Independence
4.	Security and Privacy	Naming, Location Independence, Receiver Driven, Content-based Security
5.	Heterogeneity and Interoperability	Naming and Name Resolution (Coupled and Decoupled mode), Strategy Layer
6.	Data Availability	In-Network Caching
7.	Energy Efficiency	In-Network Caching, Naming

hand-off of a mobile device with nearest new router. Security and privacy in IoTs can be provided through following features of ICN, for example ICN named contents make it easy to inspect that data is flowing according to query, content location independence hides the source of content, receiver-driven communication style confirms that content is arrived because receiver has requested for this content and self-certified contents ensures that the contents are same as sent by source. Heterogeneity among IoT devices can be handled easily when devices are named through ICN naming. Different types of IoT devices can operate with each other more efficiently when ICN strategy layer will be induced in IoT devices. ICN in-network caching can enable IoT networks to cache fetched data in (all intermediate) node(s) to enhance data availability in IoT network. Moreover, in-network caching decreases the frequency of fetching data from producer and thus saving network life and making it more energy efficient. Table 2.2 summarizes the mapping of IoT requirements to supporting ICN features.

2.2.4 Feasibility of ICN Models and Projects for IoTs

This section presents nine famous ICN architectures such as DONA [112], NDN, COMET, PURSUIT, SAIL, CONVERGENCE, MobilityFirst, C-DAX [31] and Green ICN [32]. ICN major projects and architectures along with funding sources are presented in Fig. 2.4 and are briefly discussed here. Further details can be found in

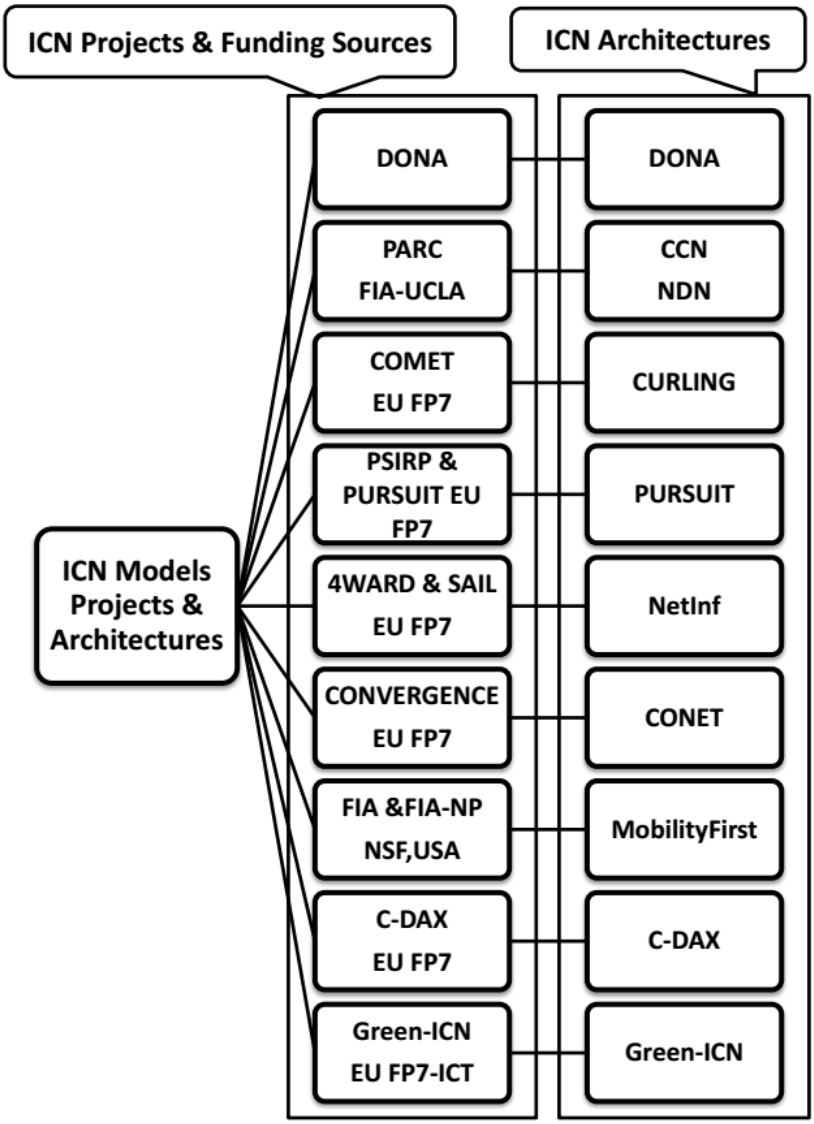

Figure 2.4: ICN Projects, Funding Sources and Architectures

[35].

2.2.4.1 DONA (Data Oriented Network Architecture)

The DONA [112] is first clean-slate network architecture in which information is named using flat naming technique and these names are self-certifying. It provides caching through both coupled and decoupled mode. Mobile subscribers get data when it subscribe network by sending Find message with a specific infrastructure and after a handoff it re-register to a new resolution handlers (RH). Mobile Publishers re-register new information after a handoff with its new RH. Due to its flat naming and undefined

methods for mobility, DONA is not feasible for billions of IoT heterogeneous devices.

2.2.4.2 NDN (Named Data Networking)

The NDN [25] is an active project from USA formerly called CCN (Content Centric Networking) [113] architecture that was a project from PARC. NDN is refining and maintaining CCN for Future Internet foundation.

In NDN, every content router (CR) maintains three data structures: i) Pending Interest Table (PIT), ii) Forwarding Information Base (FIB) and iii) Content Store (CS). PIT stores the interests that are not satisfied, FIB keeps track of named data and their corresponding data sources to forward interest and CS caches the sent data along with its name to serve in future. NDN also provides both on-path and off-path caching. Mobile subscribers issues new Interest Message for missing data while mobile publishers update their FIB entries through routing protocols. Security is provided by adding security information in the Data Packet. Due to dynamic and hierarchical naming, NDN is suitable for IoT applications. Moreover, NDN provides easy network administration, guard network against DoS attacks and spanning trees. Therefore, NDN is being employed in IoT applications like VANETs [114], smart home [36] and building management system [115]-[116].

2.2.4.3 COMET (COntent Mediator architecture for content aware nETworks)

The COMET [28] is designed to provide easy access of contents and fast dissemination of contents through content-aware networks.

COMET names contents by two readable components; its publisher name and content name signed by publisher pubic key. COMET support both on-path and off-path caching. Mobile users gets help from location aware routers which predict future location and provides data through proactive handover. However, COMET is less suitable for specific IoT applications but more suitable in data dissemination due to probabilistic caching.

2.2.4.4 PURSUIT (Publish Subscribe Internet Technology)

PSIRP [117], latterly known as PURSUIT [26] project. PURSUIT names the data through self-certifying names specifying scope and rendezvous identities. PURSUIT

offers caching in both on-path and off-path modes. Mobile subscriber is supported through caching and multicasting while producer mobility is not defined.

Like COMET, PURSUIT is also less suitable to any specific IoT application rather it can be used for data dissemination.

2.2.4.5 SAIL (Scalable & Adaptive Internet soLutions)

Both projects SAIL [118] and 4WARD [119] have shaped an ICN-based architecture named network of information (NetInf) [27]. SAIL architecture combines the naming techniques of NDN and PURSUIT. Flat names are used to find a matching publication for required data like PURSUIT. A flat name consists of a pair of authority and local. These two parts can be either hash or self-certifying components. If publication matches then long hierarchical names are used by routers to locate publisher like NDN. SAIL can operate in both decoupled and coupled fashion.

Like COMET and PURSUIT, SAIL also employed for fast data dissemination rather than any specific IoT application.

2.2.4.6 CONVERGENCE

The CONVERGENCE [29] project partially depends on existing IP-based architecture. CONVERGENCE has targeted mainly multimedia contents. A versatile digital item (VDI) based on MPEG-21, is defined as a common container for each digital multimedia content.

CONVERGENCE names the contents using both hierarchical or flat self-certifying naming mechanisms. Due to coupled approach, CONVERGENCE supports on-path caching. Mobile subscriber are handled like NDN mobile subscriber are handled while publisher mobility is not defined properly. Therefore, CONVERGENCE combines the features of NDN with its own features to use IP-based architecture. It is an overlay approach rather than purely ICN-based. In CONVERGENCE each individual packet is scanned only at the subscriber. For this reason, it may make DoS (major security concern of current architecture) possible.

The CONVERGENCE project, like SAIL, PURSUIT and COMET, is also less suitable for IoT applications and can be employed for general information dissemination and for easy management of multimedia contents.

2.2.4.7 MF (MobilityFirst)

The MF project [30] was carried out in 2010-2014 with the aim to provide a clean-slate architecture for mobile devices. Under MF project (funded under FIA), it is aimed to design and evaluate mobile data, IoT use cases and mobile entities. FIA-Next Phase (FIA-NP) also plans to unify Future Internet while employing advanced cellular networks like 4G and 5G.

In MF names are assigned and translated via global naming resolution service (GNRS). Whenever a device has data it contacts GNRS to get a name for it. This flat name is 160-bit long and known as globally unique identifier (GUID). Every network device must has its own GUID and GUIDs of the all services it produces. By naming network devices both host-based and name-based communication is possible.

MF supports on-path caching and can support off-path caching at the cost of extra registrations. Host (both publisher and subscriber) mobility is handled by GNRS. GNRS updates all references and can be accessed via accessing GNRS. Contents are flat that can be self-certifying.

MF is designed to support mobile devices and is employed in IoT applications like mobile networks, cellular networks and bus management systems [115].

2.2.4.8 C-DAX (Cyber-secure Data And Control Cloud for Power Grids)

As obvious from its name, ICN-based architecture under the C-DAX project is specifically designed for smart grids. ICN-based cyber secure communication middle-ware is designed for smart grids. Information is named and arranged in the form of topics. C-DAX works in decoupled mode to make transparent communication possible. C-DAX can also work with smart grid protocols. By employing ICN concepts, it provides cyber secure, resilient and inter-domain communication with three modes: streaming, query and point-to-point. Moreover C-DAX employs on-path caching.

C-DAX works for IoT smart grid application and electric vehicles.

2.2.4.9 G-ICN (Green ICN)

The G-ICN aims to provide ICN-based energy efficient communication solutions for disaster management (situations like flood and earthquake [120]) and video delivery. G-ICN employs ICN (specifically NDN) features like decoupled communication and

in-network caching to enable services for disaster management. Fast data dissemination of multimedia contents, is enabled through the use of in-network caching and by naming the contents. Both naming schemes e.g., flat self-certifying and long hierarchical, are employed to manage information in the form of topics. Moreover, fast and easy delivery is made possible by practicing naming and in-network user assisted caching.

The G-ICN is definitely necessary for IoT applications like efficient delivery of multimedia contents in VANETs and in disaster management as reliable communication solution.

However, their feasibility for IoTs w.r.t naming, caching, security and mobility support is discussed below and summarized in Table 2.3.

2.3 ICN-based IoT Architectures

In this section, we present ICN-based IoT research efforts which proposed ICN-IoT network architecture to support IoT needs. The purpose of mentioning these efforts here, is not to compare these in any perspective but to showcase the efficient applicability of ICN for IoT along with fertility of this research era.

To build IoT on the basis of ICN, research community is tyring hard. In this context, to support clean-slate architecture of ICN for IoTs, NDN-based high level **node architecture** is proposed in [23]. Three layers NDN-IoT architecture, consisting of application layer, NDN layer and thing layer, is presented. Node architecture includes content chunks instead of IP address enabling name-based networking. Strategy layer is introduced to provide transport and forwarding tasks according to access technologies and application needs. NDN operates at the network layer and performs its duty with the help of two planes namely control and management plane and data plane. Control and management plane perform the task like routing, configuration and service models while data plane handles interest and data messages and related jobs like strategy caching. In Fig. 2.5 we present the evolution of Internet architectures. It shows IP-based architecture, dedicated version for IoT on the basis of IPv6, extended version (to support IPv4, IPv6 and 6LowPANs) and ICN-based (NDN) architecture. To support IoT **push operations**, three different strategy schemes are presented to

Table 2.3: ICN Projects, Corresponding Architectures and their Feasibility for IoT

Project Name, Duration and Funding Source	**ICN Arch. Name**	**1.Naming, 2.Caching, 3.Security and 4. Mobility**	**Extent of Suitability for IoT App.**
DONA 2007 UC Berkeley	DONA	1. Uses flat self-certifying names, that cannot provide scalability. 2. DONA offers both on-path and off-path caching. 3. Self-certifying flat names 4. Early-binding approach	Not suitable as flat names cannot manage IoT billions of devices data contents
CCN (2010-2013) by PARC, NDN by NSF and UCLA	NDN	1. Provide hierarchical, static and dynamic named data through easy administration. 2. NDN offers both on-path and off-path caching (cache everything) 3. Publisher signature with PKI 4. Listen First Broadcast Later (LFBL)	Highly suitable as IoT devices are constraint oriented, and needs scalable naming technique
COMET (2010-2012) EU Framework 7 Programme	CURLING	1. Unspecified naming scheme, enhance easy access and fast data dissemination through content aware networks, especially supports flash crowds. 2. Works on both on-path and off path through prob-caching). 3. Public key cryptography 4. Specialized mobility-aware Content-aware Routers (CaRs)	Not suitable for IoT as naming scheme is not defined but suitable for data dissemination applications
PSIRP and PURSUIT (2010-2013) EU Framework 7 Programme	PURSUIT	1. Flat naming provides a decoupled architecture that separates name resolution and data forwarding. 2. Provides effective off-path caching 3. Self-certifying flat names 4. Facilitated by multicast and caching	Not suitable as flat naming scheme cannot manage billions of IoT devices and data contents but suitable for data dissemination applications

Project Name, Duration and Funding Source	ICN Arch. Name	1.Naming, 2.Caching, 3.Security and 4. Mobility	Extent of Suitability for IoT App.
4WARD (2008-2010) and SAIL (2010-2013) EU Framework 7 Programme	NetInf	1. Flat self certifying or hashed naming divides the whole operation in two-steps: name resolution by NRS and data routing by node itself. 2. It offers both on-path and off-path caching 3. Self-certifying flat names with possible explicit aggregation 4. Late Name Binding (LNB)	Not suitable as flat naming scheme cannot manage billions of IoT devices and data contents but suitable for data dissemination applications
CONVERGENCE (2010-2013) EU Framework 7 Programme	CONET	1. Both (hierarchical and flat Naming) schemes, converges to NDN and DONA in some aspects, designed for multimedia contents, partially dependent on IP-based architecture and partially on ICN-based, 2. Both on-path and off-path caching is provided 3. Publisher signature with PKI 4. Same as NDN with the difference at forwarding information at Border Nodes (BNs)	Not suitable as IoT application requires more than the management of only multimedia contents. IoTs architecture also needs to manage simple contents. But it is suitable for data dissemination applications.
MobilityFirst FIA (2010-2014) and FIA-NP (2014-to date) NSF, USA	Mobility First (MF)	1. MF uses flat, self-certifying naming scheme, 160-bit long names to avoid collision and make comparison easy and fast. MF provides best mobility services and employs IP-based architecture in an efficient way 2. MF offers on-path caching 3. Self-certifying flat names 4. Consumer mobility handled using Global Name Resolution Service (GNRS) and Border Gateway Protocol (BGP) for inter-domain routing	Highly required by IoT as it can have both mobile and static devices.

Project Name, Duration and Funding Source	ICN Arch. Name	1.Naming, 2.Caching, 3.Security and 4. Mobility	Extent of Suitability for IoT App.
C-DAX FP7-ICT (2012-2016)	C-DAX	1. Information is managed in the form of topics using flat and attributes-based naming	For cyber-secure smart-grids and electric vehicles.
Green ICN (2013-2016) EU Framework 7 Programme	Green ICN G-ICN	1. Contents can be named by using both flat, self-certifying and hierarchical naming schemes with attributes and arranged in topics 2. User assisted caching is employed	Highly required by IoT disaster management and multimedia contents dissemination applications

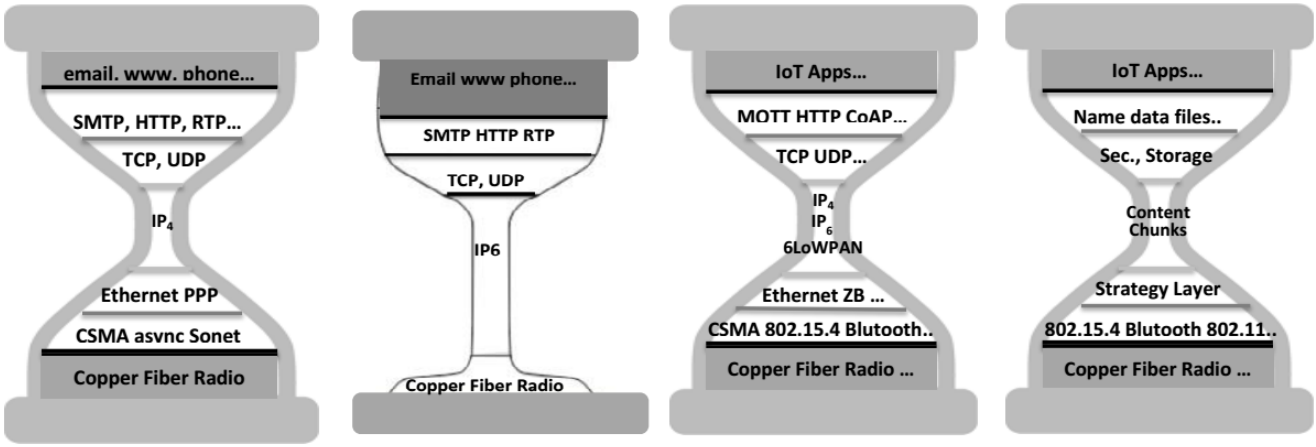

Figure 2.5: IP-based Network Architectures and ICN-based IoT Network Architecture

provide push-type communication for NDN in [121]. Natively NDN supports pull-based communication, so to provide NDN-based IoT, they provided push support in NDN. First scheme *Interest notification*, modifies interest message by including small data need to be transmitted. This small data is not meant to be cached. Second scheme *Unsolicited data*, transmits small packet of uData that is not feasible for routing. In third scheme *virtual interest polling (VIP)*, receiver transmits long live Interests such that whenever data is available, producer replies and on the failure consumer can re-transmit Interest again. They presented the analytical model for Interest notification, Unsolicited data and VIP and implemented the model in MatLab. VIP outperformed in terms of network resources used and is suitable for massive IoT

environment while other two techniques are suitable for situations where battery is critical source. Furthermore, to provide IoT **scalability**, CCN (NDN) is identified as the best candidate for IoT rather than RPL/UDP (in IPv6-based 6LowPANs) and implemented in RIOT OS through simulations [109]. Wild deployment of ICN is carried though 60 nodes located in several rooms of several buildings. CCN lightweight version, CCN-lite is simulated and they enhanced CCN through two proposed routing flavors (vanilla interest flooding (VIF) and reactive optimistic name-based routing (RONR)). Both VIF and RONR are evaluated to show that these protocols reduce routing overhead for constraint oriented devices. They also addressed positive impact of caching and naming the data. Moreover, NDN-based **secured architecture** (in Python language and Javascripting-based browser to visualize the data) is explored to secure a building and it is installed in UCLA (University of California at Los Angeles)[122]. Name-based and encrytion-based access control method is proposed and implemented to secure sensitive data. This is a initial prototype to showcase the scalability and security performance achieved by NDN instead of IP-based security systems. To address and target IoT **heterogeneity** in terms of both static and mobile devices, an unified ICN-based IoT platform is disscussed in [115]. NDN and MF are selected to cater both static and mobile devices. They provided comparison between/among both NDN and MF through building management and bus management system scenarios. Different sensors and actuator are considered as static devices while buses are considered as mobile devices. They argue and found that MF outperforms NDN when mobile objects like buses are involved while NDN outperforms MF only when static devices are involved. They have implemented NDN and MF in NS3. In [123], authors discuss NDN for smart cities and present relevant challenges.

2.4 ICN-IoT Naming Schemes

Fundamentally IP-based Internet was designed to communicate between academic devices, but with time, Internet usage has expanded from academic communication to fulfill society communication needs. Later on, as well as currently, with the help of add-on and specific purpose patches, IP-based Internet tried to fulfill current needs of society. As a consequence, by adding patches, IP-based Internet architecture provides current needs at the cost of more complex, extra expensive, delayed communication

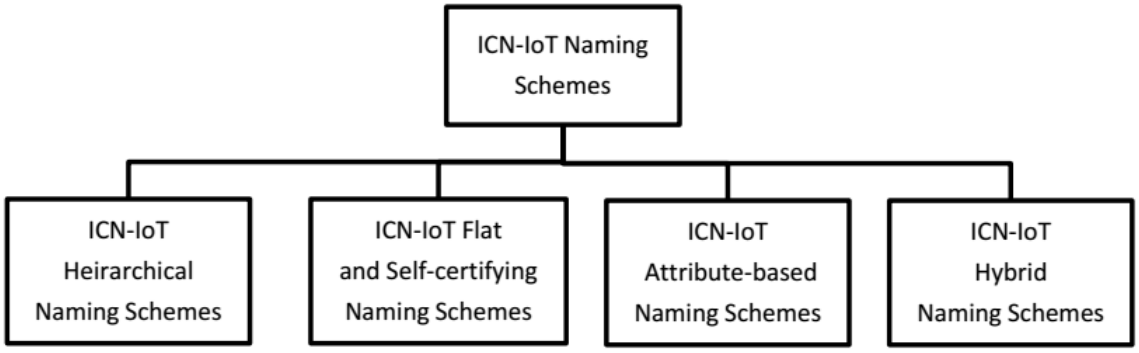

Figure 2.6: ICN-IoT Naming is Categorized into Four Categories: ICN-IoT Hierarchical Naming Schemes, ICN-IoT Flat and Self-Certifying Naming Schemes, ICN-IoT Attribute-based Naming Schemes and ICN-IoT Hybrid Naming Schemes

and sharing of content. With the time and keeping current expectations from Internet in mind, researchers proposed the idea of ICN that is based on name-based networking. The named content can be accessed independently irrespective of its location of existence. In ICN, the name of content requested is required instead of sender and receiver address pair. Therefore, this makes ICN as receiver-driven communication model in which receiver is responsible and have full control over whole communication instead of sender. Network is responsible for and will have to look for content providing best source [35]-[34].

As users are more and more interested in getting content rather than the location of the content from where it is coming, ICN approaches provides the ways to name data according to some constraints. User can get requested contents by only providing their names.

ICN naming can also outperform in naming IoTs contents. IoTs contents are transient in nature and it is undoubtedly possible for one content to have many versions based on time and sensors that generate same information. Moreover IoTs contents are huge in number like billion of billions contents are likely expected to produce in any single second and IP-based Internet cannot address 50 Billion [124] connected devices efficiently. According to CISCO report, there will be 12.2 Billion IoTs smart and constraint-oriented connected devices in 2020 [125]. In addition, IoT network architecture is assumed to support scalability and heterogeneity. Therefore, naming IoT (devices and) contents through ICN ensure, efficient addressing and scalability, more security, better mobility and support for heterogeneous devices [126]-[127].

Mainly there are two naming techniques (hierarchical naming structure and flat/hash naming) that are available through ICN architectures. CCN [128] / NDN [129] name contents in hierarchical manner while other ICN approaches (DONA [112], PURSUIT [117], COMET [28], MobilityFirst [30], SAIL [118] and CONVERGENCE [29]) follow flat self-certifying names. Third naming scheme, attribute-based has been used initially in CBCB (Combined Broadcast and Content Based) routing [130] and can be used in combination with prior two naming techniques [131]-[132]. However, most of the research efforts considered and explored hierarchical naming technique for IoTs [133]-[134]-[110]-[109]-[116]-[36]-[127]. Some researchers focus on hybrid naming schemes incorporating both hierarchical and flat with attribute-based naming [135]-[114]. We categorize ICN-based IoT Naming Schemes into four schemes which can be seen in Fig. 2.6 and summarized in Table 2.4.

2.4.1 Hierarchical-based ICN-IoT Naming

These names are human readable names and offer name aggregation. Hierarchical naming is used in NDN and CCN approaches. It follows the hierarchical structure to name contents like contents are named on web pages through URLs. Hierarchical naming provides good compatibility with the existing Internet applications and supports name aggregation. Through variable length, hierarchical names are highly scalable that fulfills the ultimate requirement of IoT contents and devices that are huge in number. Searching for a specific name through hierarchical naming already has good compatibility with existing web-browsers architectures. Hierarchical names reduces the routing table information through name aggregation[131]-[132].

On the other hand, long and variable length hierarchical names cause degradation in search efficiency and for low power devices it could create more performance degradation.

In [110]-[136] hierarchical content naming scheme is used to provide naming of contents. This work was conducted to design, implement, and integrate a CCN communication layer in Contiki based on named data for wireless sensors and networking embedded systems. A CCN name is hierarchical name attributed to content. It simply consists of a series of components of arbitrary lengths. No limitations are imposed that what sequences of byte will be used. The implemented communication

Table 2.4: ICN-based IoT Naming Schemes are Summarized According to the Fig. 2.6. Here NLAPB is for Name Look-up Solution with Adaptive Prefix Bloom Filter.

Ref.	Arch. and IoT App.	Comparison	Parameters Evaluated	Simulator (OS, Programming Platform, Lang.)
Hierarchical Naming Schemes for ICN-IoT				
[136] [110]	CCNx and Temperature Measurement Wireless Sensor Networks	IP	1.Retrieval Delay with and without caching 2.Number of Exchanged Messages	Contiki OS and Cooja Simulator
[109]	CCNx and Building Automation	6LoWPAN /RPL/UDP Vanilla Interest Flooding (VIF) VS. Reactive Optimistic Name-based Routing (RONR)	Number of Consumers VS. Number of Messages Sent (With and without Caching)	RIOT OS
[36]	NDN and Smart Home	-	1.Number of transmission(s) 2.Number of Exchanged Messages Vs Number of producers	No simulations Not mentioned
[133]	NDN and Light Control System (Instrumented Environment)	-	No simulations Not mentioned	Not mentioned
[116]	NDN and Building Management Systems	-	No simulations Not mentioned	Python-based Application Java-Scripting Data Visualization Application

Ref.	Arch. and IoT App.	Comparison	Parameters Evaluated	Simulator (OS, Programming Platform, Lang.)
Flat (and Self-Certifying) Naming Schemes for ICN-IoT				
[137]	CCNx and WSN	IP-based WSN	1.Average energy consumption 2.Average delay	Contiki OS and Cooja Simulator
[138]	ICN and Not for low-power IoT devices	Not provided	Not provided	No Simulations Not mentioned
Attribute-based Naming Schemes for ICN-IoT				
[139]	ICN and Smart Hospital	With and without ontology	1. Storage Overhead 2. Transfer Time Consumption	C Language
Hybrid Naming Schemes for ICN-IoT				
[135]	NDN and Vehicular Ad-hoc Networks	No Comparison	-	No Simulations Not mentioned
[140]	NDN and Multimedia Contents dissemination in VANETs	No naming Comparison	1.Start-up delay 2.Playback Freezing Ratio	NS3 with ndnSim
[114]	NDN and Vehicular Ad-hoc Networks	1.NLAPB 2.Simple Trie	1.Processing Time to add prefixes 2.Processing Time to delete prefixes 3.Processing Time to search prefixes 4.Memory consumption	Not mentioned
[141]	CCN and IoT Smart Campus	Hierarchical and flat naming aggregation	1.Interest transmission rate 2. Number of covered hops and exchanged messages	Contiki OS and Cooja Simulator

layer specifies only the name structure and does not assign any meanings to names. It is up to applications or global naming conventions to set and interpret meanings given to names. Application developers are free to design their own custom naming conventions. However interest is processed in a hierarchical way. Matching is performed on prefix to provide multiple responses. They used CCN for every node. Contiki OS is used with Cooja simulator to simulate physical TelosB [142] nodes. It is the first paper that implemented CCN in Contiki OS. However, only one sink (consumer) node is considered with ten to forty sensors (producer) nodes. Only static nodes are considered. Moreover, provided naming scheme is not easy to compare for a specific data as hierarchical names are long and complex to perform matching. It is suitable for IoT application having sensors deployed at fixed places (e.g., Building automation and management).

Similarly, in [36] NDN hierarchical naming scheme is modified for smart homes. Authors have provided name space specific to home related tasks. Naming scheme is designed to consist of two part: first for "*configuration and initialization*" for the smart home application and described by prefix "*/homeID/conf/*" while second part is for the "*tasks*" that need to be performed by smart home application and indicated through prefix "*/homeID/task/*". Tasks are further specified by two named-components, type (is selected from "*/action*" and */sensing*) and sub type (is chosen from real tasks like "*/light, /temp, /airCond*") respectively. Name aggregation is suggested to support task aggregation to reduce number of sent messages and hence to reduce network bandwidth. But they did not provide any simulations to show how names are carried by interest and data messages. Proposed naming scheme is designed for home scenario and thus cannot be used for other IoT applications that involve mobile devices.

NDN hierarchical naming is explored and deployed for lighting automation by UCLA [133]. Contents are named according to three parts:

/constant-namespace/command/randomizer$\|$*auth-tag*.

For instance, in "*UetTaxila/CPED/VipLab/Light01/ON/13:15:046FHDK*", here "*UetTaxila/CPED/VipLab/Light01/*" represents light numbered as "01", located in Video and Image Processing Laboratory (VipLab) in Computer Engineering Department (CPED) of University of Engineering & Technology, Taxila (UetTaxila),

"*/ON/*" directs to turn this light "ON" and "*/13:15:046FHDK*" indicates the time and corresponding computed hash of the name to ensure security of the content.

Authors in [109] have implemented NDN on IoT constraint-oriented devices for building automation. They have demonstrated the use of small names of size up to 12 bytes. They find NDN can support maximum name length up to 30 bytes. They believe that hierarchical, short and non-human-readable names are highly suitable for IoT smart devices while maintaining name-aggregation.

While in [116] authors believe hierarchical, human-readable names and application-specific names simplify both creation and processing tasks. NDN naming scheme is implemented to secure using ICN for UCLA campus. Designed prototype is implemented in Python and embedded in a browser-based interface. Namespace comprised of main root name followed by two sub-category names. For example, "*/ndn/ucla.edu/bms/building/strathmore/data/power/<time-stamp>*" specifies NDN application deployed at UCLA university for university-building-management-system and fetches power data according to specified time of strathmore building located in UCLA. Moreover other sub-name space,"*/ndn/ucla.edu/bms/user/public/key/<key-id>*" directs NDN-based BMS application towards public user (having multiple keys) through user specific key.

However, we argue that short-hierarchical names are suitable for IoT contents because it offers high scalability and name aggregation. Therefore, researchers need to look for the solutions to improve look-up efficiency and optimization of routing table size for IoT constraint oriented devices.

2.4.2 Flat-based Self-certifying ICN-IoT Naming

ICN native approaches like DONA [112], MobilityFirst [30] and *NetInf* [27] follows flat, short and self-certifying names. These names can be computed using the hash of content or of any part of it and thus can be non-human-readable. Flat names can be of any fixed length and therefore simple and easy to process in routing as it take less computing resources, and consume less space while saving.

Although there are very few research attempts that explored ICN flat naming alone. We survey and present these flat naming research efforts in following paragraphs. Moreover, these efforts are not for IoTs.

In [137], authors presented ICN flat naming scheme for WSNs. Presented naming scheme have two parts: first is to identify category and second is for content. They have investigated CCN naming in Contiki OS and results indicate that proposed naming scheme outperform IP in energy consumption and delay.

In [138] authors present routing scheme based on flat naming. To provide name aggregation and efficient searching, bloom filters are used. They have introduced the concept of containers to save contents. Containers are controlled by controllers and accessed through access controllers. Flat names play a great part in routing of contents because they are short in length and this makes it easy and less complex in comparison. However, this work has not involved constraints required by low-power constraint-oriented devices, and hence, is not suitable for IoT applications.

In [143], authors survey ICN architecture naming schemes and argue that self-certifying names provide name-persistence, security-binding and universal uniqueness. Moreover, in [144] naming schemes comparison is provided and authors argue that flat names are agnostic to the structure of the data, easy to manage and seems more scalable at the network layer. Most of the work regarding flat names is conducted for name base routing[145]-[146].

However, on the other hand, flat names does not provide name-aggregation which is needed for IoT contents and devices to ensure scalability. Thus, flat names can increase the routing table entries making it complex. It will increase delay to process a query and will need large space. Moreover, most of the flat names are non-human-readable, therefore to respond any query, a third-party translation mechanism will be required. IoT devices are small in memory and power, so flat names alone are not suitable for IoT contents and devices.

2.4.3 Attribute-based ICN-IoT Naming

This naming approach extract attributes of content and was used initially in CBCB [130]. This naming approach does not ensure global uniqueness of the content. Content attributes can include production date and time, content type, content location, content version number and any specific property of the content etc. Therefore, attribute-based naming support searching using easy and known key words for the

content. Although it is obviously possible to find many responses against single query and its hard to find unique content in short time.

To secure contents, a routing scheme is provided in [147] using attributes of the content. In [139], attribute-based naming scheme is presented with the help of ontologies to manage contents in distributed environments. Authors claim that proposed attribute-based naming scheme provide better privacy, simple namespace management and reduces computation cost for user to determine accessibility. A hospital scenario is presented and described. In our observation this attribute-based accompanied ontologies naming scheme can outperform in IoT applications where privacy is highly needed, for example smart-health and smart-transport.

In [143], authors believe and suggest to use keywords of content created by owner as they take less time in searching while making lookup process easy.

For IoT applications, attribute-based naming can help in a perspective that IoT applications are extremely different and user can specify required content name in keywords. Attributes can be saved as keyword or hash of attributes to provide more security. Efficient advance search is only possible through attributes of the content. However, fetching unique content seems difficult with only attribute-based naming. To make this happen, other naming schemes can be combined in a hybrid fashion.

2.4.4 Hybrid ICN-IoT Naming

Hybrid ICN-based naming schemes for IoTs, refer to naming schemes combining three naming schemes or any two of them. The purpose behind combining above mentioned three naming schemes is to utilize their best features for IoT applications. Advantages of these hybrid naming schemes are manifold like improved security, better compatibility, enhanced scalability and easy name management [131]-[132].

In [135] scalable naming scheme is proposed for mobile nodes like vehicles and their produced mobile contents. Content name consists of three components:

i) Scheme, "*vhn*" which specifies the vehicular network or vehicular identifier,

ii) Prefix that is actually a hierarchical component, that contains information of producer (car) and details about content, and

iii) Flat part is the hash of the item, owner or signature of owner.

However, they did not provide any supporting simulations and feasibility for the proposed scheme. Moreover, the proposed naming scheme based names can be very long and suitable for VANETs only. This scheme is complex for IoT constraint-oriented devices as they can hardly forward/store such long names from/in their CSs.

In [140], hybrid naming scheme is proposed and used for multimedia contents in VANETs using ICN. Proposed naming scheme comprised of following three parts:

i) Prefix "*hmn*": indicates "hierarchical multimedia naming" and hierarchical component names and used for routing and name-aggregation ,

ii) Flat part is the hash computed on complete name or part of it and

iii) Attribute part is the attributes of the content.

These three parts (prefix, flat and attribute) are separated by ":" while both prefix and attribute sub-components are separated through "/". This work is designed and evaluated for the dissemination of multimedia contents in VANETs.

In [114], authors investigated hybrid naming scheme proposed in [135] and presented their corresponding results for VANETS. Authors claimed that proposed hybrid naming scheme take less space to save more names as compared to NLAPB [148] and simple trie. They have performed simulations and results indicate that lookup time and memory management improves for VICN. Maximum prefix allowed length counted as 72 bytes. Therefore, this hybrid naming scheme is well suited for low power devices and can support IoT devices when underlying technology is IEEE 802.15.4 Zigbee [149] (i.e., Payload size is 127 Bytes).

In [141], we proposed hybrid naming scheme for IoT-based Smart Campus (IoTSC). Hybrid naming scheme names the IoT contents while combining hierarchical and flat components. Proposed naming scheme takes domain name, location, task as hierarchical component and hash of device name as flat component. Flat component is computed through FNV-1a hash. Through hashing, integrity of content is maintained. Proposed scheme is evaluated and simulated for Zigbee both static and mobile devices in Contiki OS with Cooja simulator. Results shows the better performance is achieved in terms of interest satisfaction rate, number of covered hops and name-aggregation.

Through ICN-based hybrid naming, many advantages of the above described schemes (hierarchical, flat and attribute) are expected to improve further while minimizing the effects of drop-acts in case of IoTs.

2.5 Chapter Summary and Insights

In this chapter, we discussed and presented related literature of both new paradigms IoTs and ICN. We elaborated briefly IoTs four working phases namely: acquisition and sensing, data transmission, data processing and information management, action and utilization along with their corresponding technologies. Requirements and challenges to build a reliable and inter-operable communication network architecture for IoTs are presented. Through this chapter, we have also discussed ICN suitable features, different ICN projects for the future Internet design and their resulting ICN based network architectures for IoTs. ICN projects are briefly discussed in terms of their corresponding feasibility for IoTs in terms of naming schemes, caching mechanisms, security and mobility support. Mapping of IoTs communication network architecture requirements against ICN striking and supporting features is presented. Furthermore, we discussed ICN based solutions/architectures for IoTs to present the applicability of ICN for IoTs.

Then from the naming perspective, we have surveyed ICN-based naming schemes proposed and investigated for IoT applications. We categorized ICN-based naming schemes for IoT into four categories: hierarchical, flat, attribute-based and hybrid naming schemes.

Our naming literature indicate that for IoTs, NDN (CCN) hierarchical naming schemes and hybrid naming schemes gained more attention from research community as compared to flat and attribute-based naming schemes. We observe that main reasons behind NDN (CCN) hierarchical naming feasibility for IoTs are both simple and easy name-aggregation and better support for scalability. Moreover, human-readable hierarchically structured names with unlimited length provide faster searching as compared to other schemes and name-aggregation saves a lot of space while making routing easy.

On the other hand, ICN-based hybrid naming enhances the benefits of combined naming schemes. Hierarchical component is added with the aim to provide scalable and efficient name aggregation with less number of entries to make routing process simple and easy. While flat-name component is concatenated to ensure improved security and privacy. Attributes of content are included to make fuzzy searching possible through attribute keywords.

Our literature survey identified that very few research studies have adopted and investigated flat and attribute-based naming separately for IoTs. Although fixed length, non-human-readable flat naming provide better security and privacy through more easy and simple computations but they do not provide better scalability, name-management and aggregation. And this is the obvious cause behind less motivation to explore flat naming for IoTs. Though, we highly suggest to use flat names to meet IoTs privacy and security requirements as a name component.

Similarly, attribute-based naming schemes alone gained less attraction from ICN-IoT research community. Attribute-based naming can assist better in advance IoT applications (for instance, an IoT application need temperature values extracted from both node 1 and 10 during the time 04:00AM to 06:00AM for any specific date from the desired area) requiring contents according to specified features. Thus, we recommend that attribute-based naming should be explored for IoTs.

However, to conclude, we recommend that hybrid naming schemes will outperform to name IoT contents and devices accompanying hierarchical, flat and attribute-based naming.

Chapter 3

Contributions and Research Methdology

In this chapter, we present research method we adopt for this thesis. In the first section, research method is presented and then we discuss and present proposed ICN (NDN)-based IoT architecture in second section. Then research steps and contribution against every step is discussed following with summary.

3.1 Research Methodology

For a systematic process in research, a procedural approach method shall be followed in order to increase our understanding of the phenomenon about which we are interested. *"A research method is a way of investigating an empirical topic by following a set of pre-specified procedures"* [150].

In this Ph.D. thesis research work, triangulation method is used. *"In which two or more methods or techniques are used to investigate the same research question or the collecting of information from several sources about the same event or behavior"* [151].

In this research, the triangulation method as shown in Fig. 3.1 is followed with a combination of *case study of IoT-based Smart Campus (IoTSC)* and *System Development Method*. IoTSC is considered as a case study because it is a true representation of IoT environment due to the involvement of heterogeneous devices and style of their connectivity. IoTSC can contain many devices like laboratory devices, class room devices and staff devices. Moreover, management of the data contents which these devices produce, also plays part in IoTSC application. Therefore, handling of

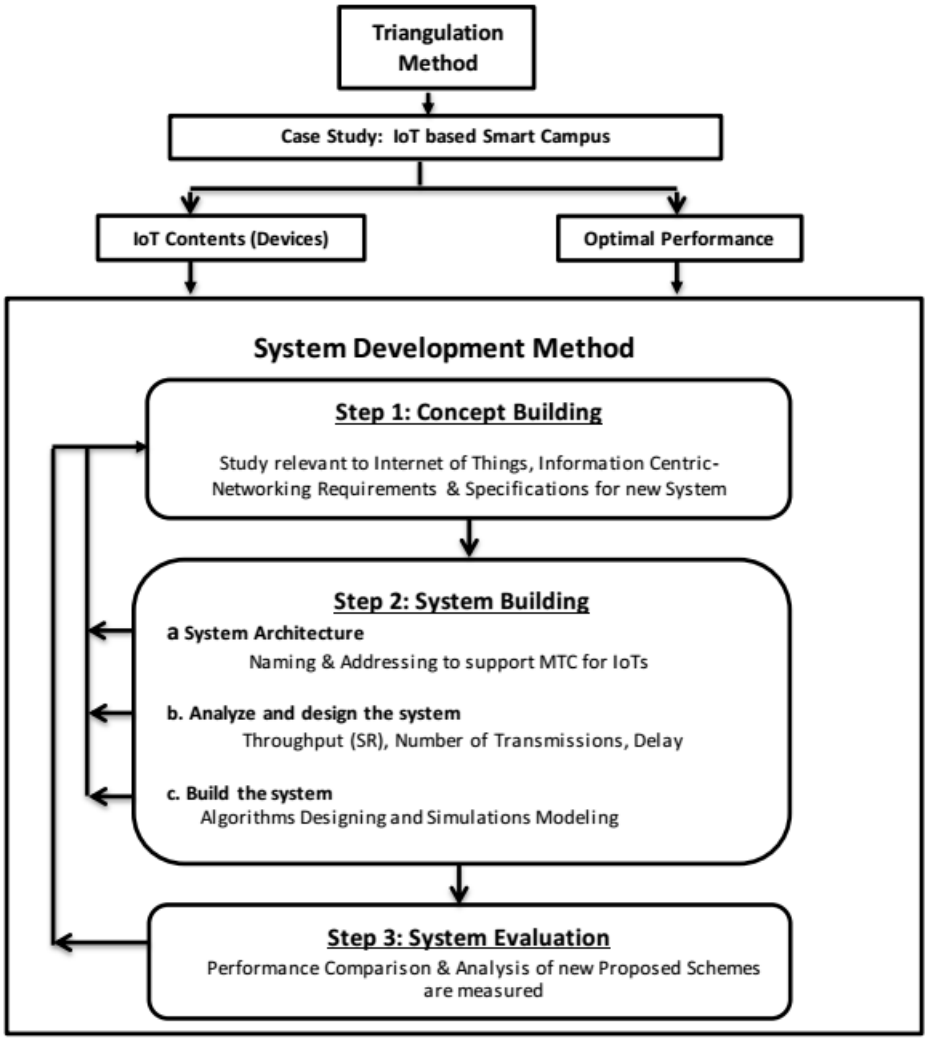

Figure 3.1: Research Methodology.

IoTSC devices and relevant contents is an important pillar of IoTSC. On the other hand, handling of devices and contents with optimal performance is also required. Thus, IoT devices and contents, IoT optimal performance are important perspectives for IoTSC case study.

Then procedural approach is made such that a ICN (NDN, CCN)-based framework is designed in which named information can be transmitted and received while considering IoTSC. An IoTSC case study is designed which included static and dynamic entities to support IoT like environment.

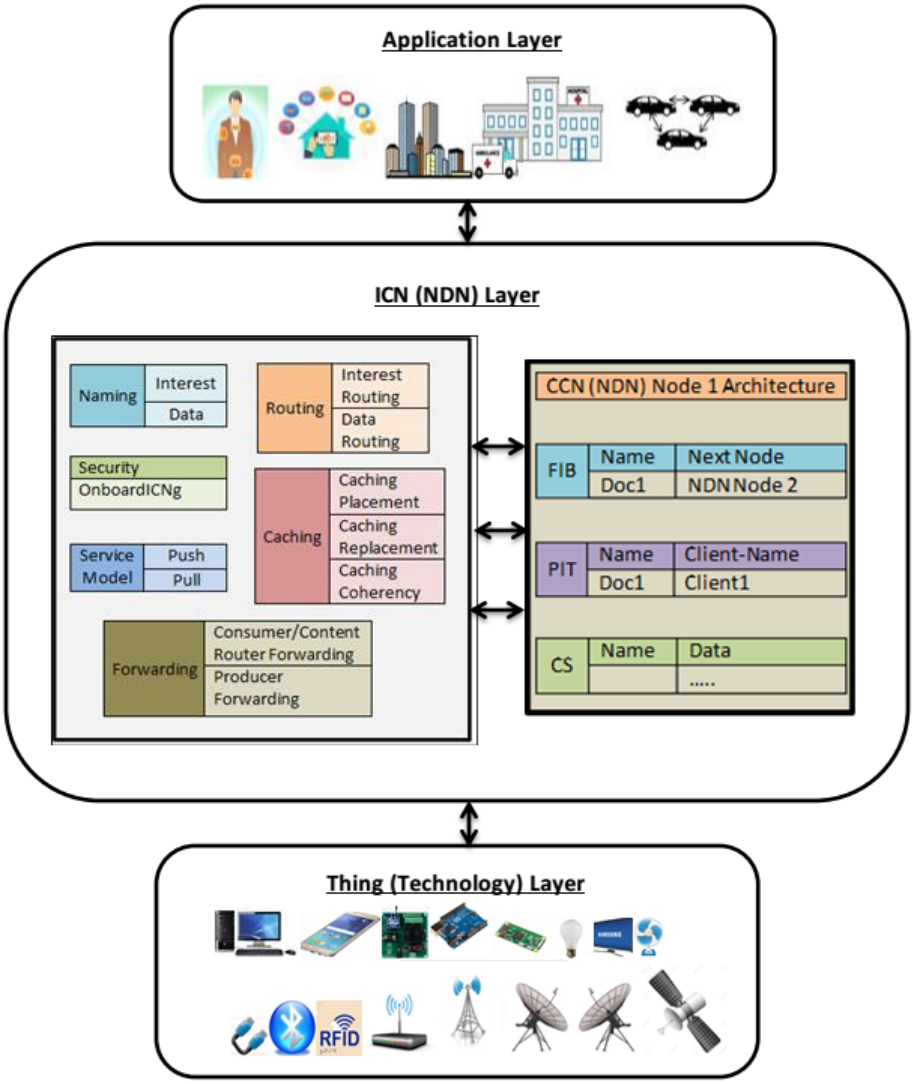

Figure 3.2: ICN(NDN)-IoT-Architecture.

3.2 Proposed ICN (NDN) for IoT

In this thesis, we introduced the ICN (more specifically NDN (CCN)) for IoT and presented two novel naming mechanisms along with optimal forwarding scheme equipped with security, heterogeneity and scalability. ICN based IoT architecture is shown in Fig. 3.2. It can be seen that ICN-IoT involves three layers. *Thing and technology layer* is aimed to address things based on underlying technologies , for example, Zigbee based light bulb and interfaced with mobile-phone. Data gathering (sensing) and actuating is performed in this layer. Then, *ICN (NDN) layer* plays the role to address naming, caching, security, forwarding, routing and provide support for service models

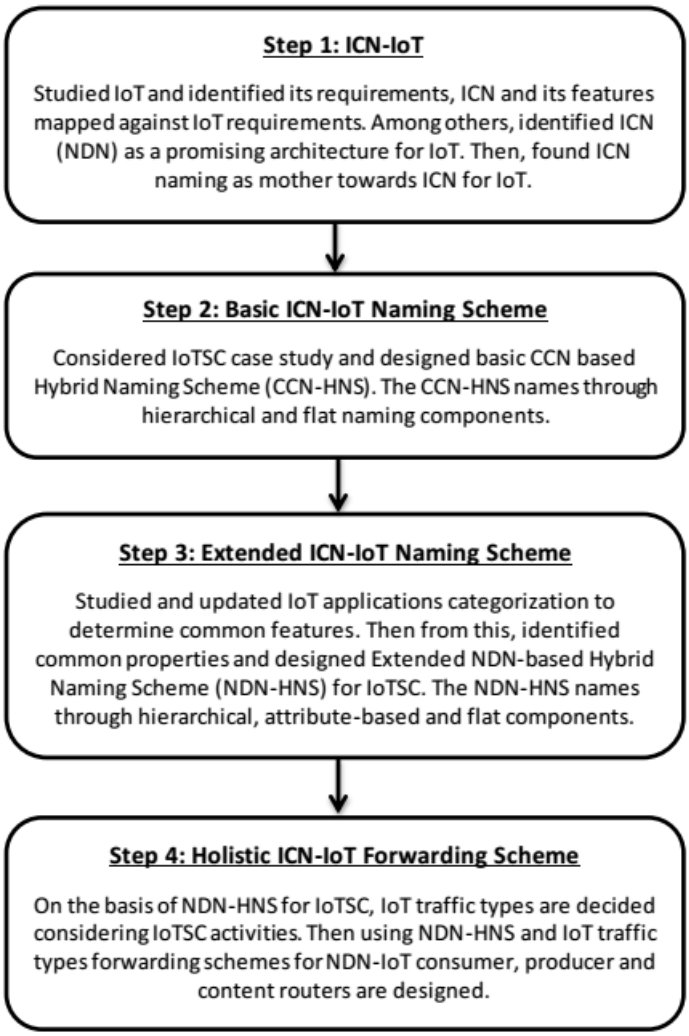

Figure 3.3: Research Methodology in Phases.

to enable MTC. *IoT application layer* connect to this NDN layer. IoT applications like smart home, smart campus, smart grid, smart city build their own interfaces on application layer like android applications to perform particular actions like harvesting of any piece of information from corresponding devices or performing some action on devices.

3.3 Research Steps and Contributions

We update IoT applications categorization into fourteen categories. From updated categorization, IoT-based Smart Campus (IoTSC) scenario is selected to design the naming and forwarding schemes due to its true representation for IoT. Moreover, brief

introduction of all important steps of methodology can be visualized in Fig. 3.3.

As a basic scheme, CCN-based hybrid naming scheme is proposed which names the contents using hierarchical and flat components to support both *push* and *pull* communication and introduced two transmission modes namely (1) unicast mode and (2) broadcast mode to address loop problem associated with CCN. The hierarchical component takes the domain name, location, task and device name in URL style. By using 'task', push support is added in the native CCN protocol. The flat component is used to provide integrity and it is computed through the FNV-1a hash of the device name and Data. The communication loop problem associated with CCN protocol is eliminated by implementing 'unicast' protocol on the source nodes. Mobile IoT nodes are used for delivery of Interest and Data packets to nodes that are not in the range of sink node. The proposed naming scheme is evaluated for IoT-SC having both static and mobile nodes.

Further as an extended scheme, a sophisticated NDN-based hybrid naming scheme is proposed which names the IoT devices and content using hierarchical, flat and attribute components to support both *push* and *pull* IoT traffic models. Proposed NDN-HNS operates in three important steps to assign a name to any content. As a first step, HC holds value about campus, content originator/requester and content itself. In second step, AC contributes content's attributes like freshness, popularity and task type. Security and authentication of the content is added by third component i.e., FC.

Then on the basis of extended NDN-based naming scheme, IoT traffic types are defined using the listed activities in IoTSC. Holistic forwarding schemes are proposed for NDN-IoT consumer, producer and content routers which provide machine type communication (MTC) with push and pull communication models enabled. These forwarding schemes use another OnboardICNg security scheme which is designed to authenticate and authorize the devices to perform asked actions. These schemes enable NDN-IoT producer to send critical content or updates of subscribed content to NDN-IoT consumer through content router(s). Moreover, NDN-IoT consumer is enable to send message to perform any action or setting value of any parameter of NDN-IoT producer.

3.4 Chapter Summary

In this chapter, we described how we used famous triangulation research method in this thesis. Then, we describe proposed ICN mainly NDN for IoT architecture. It follows with the brief overview of research steps and contributions in the form of abstract level details of this work. Specific details of steps and contributions we made can be found in coming chapters.

Chapter 4

Basic CCN-IoT Naming Scheme: Hierarchical and Flat based Hybrid Naming Scheme in Content-Centric Networks of Things

Information-Centric Networking (ICN) approaches have been considered as an alternative approach to TCP/IP. Contrary to the traditional IP, the ICN treats *content* as a first-class citizen of the entire network, where *names* are given through different naming schemes to *contents* and are used during the retrieval. Among ICN approaches, Content-Centric Networking (CCN) is one of the key protocols being explored for Internet of Things (IoT), names the contents using hierarchical naming. Moreover, CCN follows pull-based strategy and exhibits the communication loop problem because of its broadcasting mode. However, IoT requires both pull and push modes of communication with scalable and secured content names in terms of integrity. In this chapter, we propose a hybrid naming scheme that names contents using hierarchical and flat components to support both *push* and *pull* communication and to provide both scalability and security, respectively. We consider an IoT-based Smart Campus (SC) scenario and introduce two transmission modes namely (1) unicast mode and (2) broadcast mode to address loop problem associated with CCN. Simulation results demonstrate that proposed scheme significantly improves the rate of interest transmissions, number of covered hops, name aggregation, and reliability along with addressing the loop problem.

4.1 Introduction

During 2008-2009, number of connected devices exceeded the number of humans on Earth for the first time and had introduced the concept of glorious IoT [152]. With the proliferation in the number and the usage of these low-cost and low-power connected devices in the form of mobile devices, RFID tags, sensors, actuators and smart gadgets has further assisted in the transformation of IoT from concept to reality. IoT is also considered to be the main force behind the 4th industrial revolution which helps to build smart infrastructures [153]. IoT devices are equipped with sensing, processing, actuating and transmission capabilities. When these IoT devices connect to the internet along with human internet users, production of huge amount of useful data is an obvious outcome [154]. Management of these billions of IoT devices and huge amounts of the data they produce (through TCP/IP networking model) is facing many issues [4].

Before IoT era, the researchers were trying to address the issues of traditional TCP/IP model and proposing modifications to make it suitable for future internet.

Although many intermediate solutions were proposed like Content Delivery Networks (CDN) but these efforts resulted in a known concept of ICN. In addition, internet users found more interested in content from internet (i.e., it can be any host) rather than communication with any specific host. In IoT era, ICN has appeared as a promising solution of the issues of TCP/IP model. ICN consider 'content' as main element of the internet architecture which triggered several projects including DONA, Named Data Networking (NDN), CCN, MobilityFirst, CONET, PURSUIT, NetInf and CURLING [35]. Among these, CCN is a prominent proof of concept [33].

Basically, CCN is a pull-based (i.e., consumer-driven) communication model where service subscriber gets data irrespective of location of content provider (publisher) only when it is subscribed for a content. CCN communication is based on two packets *'Interest'* and *'Data'* which are used to request and transfer the content, respectively. CCN can run on any network-enabled device because of its content-centric nature. CCN routers uses three data structures namely Pending Interest Table (PIT), Forwarding Information Base (FIB) and Content Store (CS). PIT keeps track of entries that are waiting to be satisfied. CS stores a copy of requested data to ensure data availbility. FIB saves interface information against any name received or generated

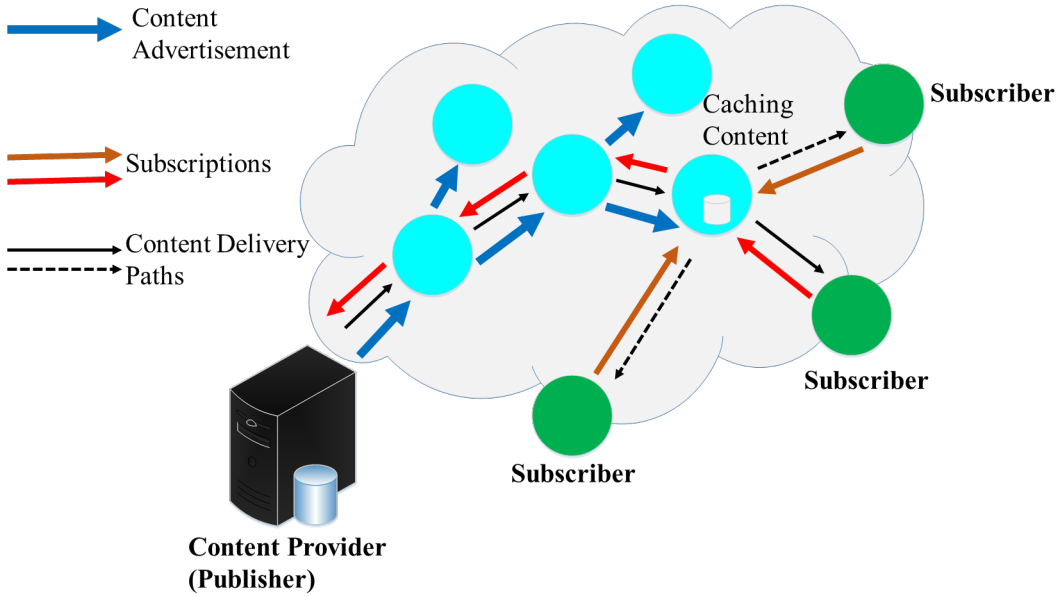

Figure 4.1: Basic Content Propagation and Retrieval in CCN.

in the Interest packet. If the Interest is not satisfied at all, there will be no entry in the FIB [155].

In CCN, content name is assigned by the publisher of the content and this name helps to cache the content at intermediate device. Fig. 4.1 shows the basic content propagation and retrieval used in CCN. There are two general CCN naming schemes (1) hierarchical and (2) flat [35].

The *hierarchical naming* is adopted in both famous CCN and NDN architectures. The hierarchical naming assigns content name in human-readable fashion using the hierarchical structure. It is similar to URL such as locating a web page with a given URL (e.g., /www.uettaxila.edu.pk/content/a.avi). "/" is a separator between the contiguous components of a name in the given example. The advantages of hierarchical naming include compatibility with current system and aggregation of data to minimize the routing information while improving the efficiency to search in a routing table. On the other hand, the disadvantage includes the variable length of the component name. Due to variable component name length, content name becomes long and it becomes very difficult to remember the long names [127]. The *flat naming* is utilized by the future network ICN architectures like NetInf, MobilityFirst and DONA [35]. Flat name is generated through cryptographic hashing of the content

or its sub-component or its attributes. This naming procedure is unique in the sense that it assures location independence, application independence and global uniqueness. However, it slows down the aggregation mechanism (just like IP addresses aggregation) which increases the size of routing tables and their entries. Furthermore, the flat names may become unreadable which requires an additional system for the classification of readable and non-readable names.

More specifically, CCN-based IoT content naming mechanism needs support for: global unique information retrieval, location independence, security, push-support and human-readability. Push support is required when any subscriber of application wants to perform any action on content producer (i.e., sensors). Thus naming the contents by incorporating above features is one of the key expectations from CCN-based architecture design for IoT [156] and the main motivation behind our work. Our proposed hybrid naming scheme combines both hierarchical and flat naming schemes from ICN models, add a sub-name (i.e., *task*) in content name to support both push and pull communication modes and appends hashed value of content to enhance security while maintaining integrity about content and its provider.

The main contributions of this chapter can be summarized as follows:

- We propose a hybrid content naming scheme for CCN that incorporate both hierarchical and flat ICN naming schemes for IoT-based SC.
- We introduce a sub part in the content name to support both pull and push style communication among IoT devices.
- We consider IoT-based SC having heterogeneous IoT nodes (i.e., both static and mobile) to evaluate proposed naming scheme and achieve significant performance in terms of interest satisfaction rate, number of transmissions and latency, name-aggregation and security (i.e., integrity) over existing schemes [36]-[114].

The rest of the chapter is organized as follows. Section 4.2 discusses the preliminaries and related work. Section 4.3 describes the proposed naming scheme while its investigation is discussed in Section 4.4. Performance evaluation along with results is discussed in Section 4.5 and we conclude the work with tentative next aspects in Section 4.6.

4.2 Preliminaries And Related Work

Here, we discuss few key motivations of considering CCN for IoT followed by the latest literature on naming schemes.

4.2.1 CCN-based IoT in a Nutshell

IoT environment have both static (e.g., sensors and machines in smart home, e-health and in smart industry) and mobile devices (i.e., connected smart vehicles). Most of IoT devices are low-power and low-cost which needs to be energy-efficient, secure, reliable, interoperable and scalable. While, in CCN, as data publisher apply naming to the content independently from its own location, this feature provides decoupling of publisher and subscriber and enables easy caching. Exploiting the CCN for IoT can provide aforementioned advantages over traditional TCP/IP architecture [156]. Furthermore, most of the IoT applications, contents and services require more, than only the communication between specific devices. As CCN disseminates contents and enables services through naming and this makes, naming the contents (i.e., instead of IoT devices) oftenly more significant. Next, with inherent caching and processing, IoT constraint-oriented devices can benefit even more due to their low-power nature. As in IoT environment, it is necessary to disseminate and retrieve content to multiple places with less number of transmissions, therefore caching and processing contents in the IoT local networks is important to save device battery and network bandwidth. In addition, during content request and response, caching helps to reduce delay in the result of local cache access. Moreover, caching the contents enhances the *data availability* where devices oftenly run out of power battery. In short, CCN approach for IoT can

1. Significantly decrease the complexity of auto-configuration methods through naming the IoT devices, contents and services in comparison to TCP/IP (i.e., a layered protocol stack) as TCP/IP protocols including DNS, CDN, TCP, routing and IP are merged in to one CCN [157].

2. Offer number of opportunities to powerfully factorize functionalities (i.e., buffering and caching for the data availability and energy efficiency [158]-[109]).

Table 4.1: Comparison of Memory footprints for Hardware and IoT Operating System

IoT Operating System: RIOT			
Module	Developing Board /Processor	ROM (KB)	RAM (KB)
CoAP+RPL+6LoWPAN	Cortex-M3	48.5	10.7
NDN		15.6	2.7
CoAP+RPL+6LoWPAN	ARM7	78.6	8.8
NDN		22.2	3.5
RPL+6LoWPAN	MSBA2	53.412	27.739
CCN-Lite		16.628	5.112
IoT Operating System: Contiki			
Module	Developing Board /Processor	ROM (KB)	RAM (KB)
CoAP+RPL+6LoWPAN	ARM7	61.3	16.5
CCN		13	5.7
RPL+6LoWPAN	RedBee-Econotag	52.131	21.057
CCNx		13.005	5.769

3. Achieve lesser memory footprint in comparison to 6LoWPAN/RPL/IPv6 [109]. In Table. 4.1, it can be clearly seen that ICN projects exhibits less memory as compared to TCP/IP-based solutions for IoT.

4.2.2 Related Work

Most of the IoT application scenarios follow naturally CCN model features. For example, IoT sensor content retrieval and mobile content updates use naming the contents and decoupling between subscriber and publisher. These IoT scenarios get further improvements through CCN in-network processing, caching and light-weight memory footprints. Actually, CCN concept and its design is more feasible for IoT wild deployments in both energy-efficient and memory-efficient manner.

The naming scheme presented in [36] employs the hierarchical naming mechanism. However, this scheme does not ensure security to critical information parts or sub parts. Furthermore, the authors consider only the static nodes which limit the applicability of their naming scheme in an environment where both static and mobile nodes are present. Syed et al. [114] introduced naming scheme for vehicular ad hoc networks. They only worked for vehicles. However, they did not consider static nodes

Table 4.2: Summary of related work: HN is for Hierarchical Naming and FN is for Flat Naming. FN is for Security.

Ref.	Objective	HN	FN	Push Support	Remarks
[36]	ICN-based naming for Smart Home	✓	✗	✓	Only Static Nodes considered and Not for hybrid IoT environment
[114]	ICN-based Naming for VANETS	✓	✓	✗	Only Mobile Nodes considered and Not for hybrid IoT environment
[122]	ICN-based Naming for smart building	✓	✗	✗	Only limited Static Nodes considered and Not for hybrid IoT environment
[159]	ICN-based Naming for under water	✓	✗	✗	Only limited Static Nodes considered and Not for hybrid IoT environment
Proposed Naming Scheme	ICN-based Naming for Smart Campus	✓	✓	✓	Suitable for IoT environment with both static and mobile nodes

and push communication style. Safdar et al. [159] implemented NDN hierarchical naming for underwater monitoring having static nodes without security and push support. Sugang et al. [115] compared both MobilityFirst and NDN models based on two scenarios (i.e., static and mobile). However, the static and mobile scenarios are addressed separately. Moreover, reagarding mobility support, names of content remains same in CCN because it supports data access through content names (rather than the IP addresses), this also helps mobile users to access the data continuously even if the published content is unavailable during the content flow. Meisel, in [160] argued that current architecture of the Internet and its protocols are not suitable for an extremely mobile environment like MANET.

In summary, all of the above presented naming schemes addressed issues with a major focus on static devices without security and push support. Summary of most relevant literature along with important comparative factors is presented in Table 4.2. Contrary to the existing naming schemes and considering the fact that IoT needs to be able to send data and to be controlled remotely, our proposed naming scheme incorporates the security and push communication aspects in the naming mechanism for both static and mobile nodes.

4.3 Proposed Naming Scheme

In this section, we describe the design of proposed hybrid naming scheme. We combine the positive attributes of both hierarchical and flat ICN naming schemes and present a novel hybrid name scheme which has two key parts: (1) hierarchical component (2) flat component. We also discuss about the processing of interest and data messages following with the tentative advantages of proposed naming scheme.

4.3.1 Hierarchical Component

This component is generated by following Uniform Resource Identifier (URI) syntax having three subcomponents: (1) Domain Name (2) Location (3) Task. Like NDN naming scheme, forward slash "/" is used as a separator between the sub-components of the hierarchical component. These all three sub-components are mixed to form hierarchical component (e.g., DomainName/Location/Task can have values of uet-taxila/CP/DC/action:on as shown in Fig. 4.2. Table 4.3 describes the definitions for these sub-components. The hierarchical component works as a prefix to a) locate the domain name b) information retrieval site or c) the location of task to be performed. Task component represents the type of task which has two types: (1) Action and (2) Sensing. This representation provides name aggregation to optimize the routing table. The entries with a matching prefix are aggregated to minimize the number of entries in the routing table. An identifier, assigned to each aggregated entry using the arrival face of that entry, differentiates them. When an Interest is satisfied, it sends the retrieved Data to all the identifiers of an aggregated entry and then removes them from the routing table.

4.3.2 Flat/Hash Component

This hierarchical component is further combined with second (flat) component using ":" as separator. The flat component deals with device name and data. It has a limited length of 32-bits, which overcomes the problem of long hierarchical names. It has low aggregation but it ensures names uniqueness. To secure the device and data generated, FNV-1a hash [161], a non-cryptographic hashing algorithm is utilized. For example, considering the Computer Engineering Department of University of

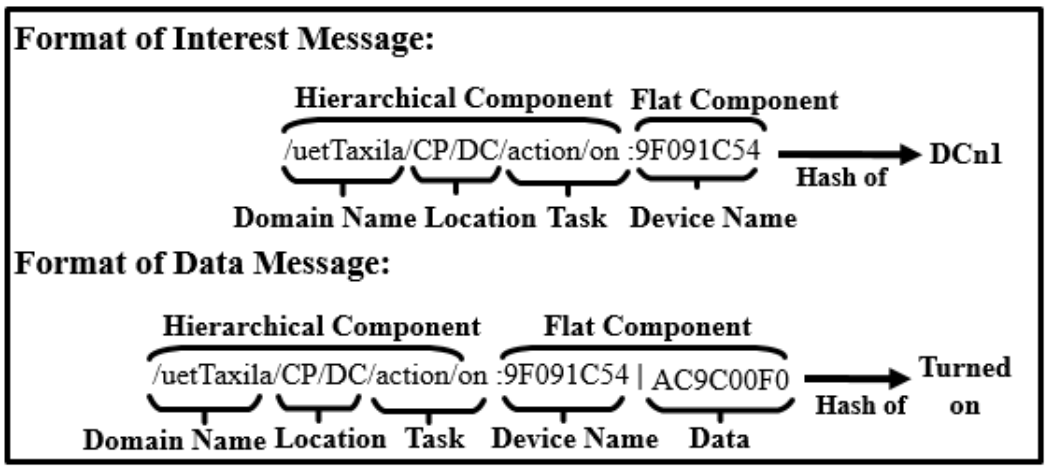

Figure 4.2: Example of Proposed Naming Scheme.

Table 4.3: Description of The Components of Proposed Naming Scheme

Components	Description
Domain Name	It is used to represent the name of the organization or building
Location	It is used to represent the location of the node where to send the Interest message and from where to retrieve the Data
Task	It is used to represent the tasks that can be performed like sensing, action, etc.
Device Name	It is the flat part which is used to represent the device uniquely by encrypting its name
Data	It is also flat part which encrypts the data to ensure its integrity

Engineering & Technology (UET) Taxila, Node1 (n1) is in Data Communication (DC) lab then the pseudo-code for the algorithm is as follows:

h = offset-factor for each octet to be hashed

h = h xor octet

h = h * prime-factor

return h

Where 'h' represents the hash value while octet represents the octet of data. Offset-factor and prime-factor depend upon the length of hash. The values of offset-factor and prime-factor for 32-bit hash are 16,777,619 (0x01000193) and 2,166,136,261 (0x811C9DC5) respectively. The hash for DCn1 is 0x9F091C54 [162].

Moreover vertical bar "|" is used as a separator between Interest Message content

Algorithm 1 Processing of Interest in the Proposed Naming Scheme
Input: Interest [Domain Name, Location, Task, Device Name]
Output: Data [Domain Name, Location, Task, Device Name, Content]

```
1: Generate Interest at Sink Node.
2: Assign Identifier 0 to PIT at Sink Node and Send Interest to
Neighboring Node(s).
3: repeat
4:    if Content Not in CS then
      if Device Name Not in PIT then
         Add [Location, Task, Device Name, Face] in PIT.
         Initialize Timer (s).
         Forward Interest using FIB.
      else
         Drop Interest.
      end if
   else
      Send Data.
   end if
5: until Data Sent or Interest forwarded to Last Node(s)
6: Send Data to requesting Intermediate Node.
```

name and Data Message content.

4.3.3 CCN-based Interest and Data Message Processing

Algorithm 1 describes the reception and forwarding of Interest message in the network. When any sink (consumer) node generates interest packet to get any named-content, this sink node adds interest with identifier 0 in its own PIT and forwards this Interest towards neighboring nodes. Then in the network, every node checks this specific interest in their CS and transfer data if find. If Interest is not found in CS then it checks in PIT and if found then it drops this Interest (It means that the request is already arrived for this Interest). If Interest entry is not found in PIT, then to add new entry of this Interest its location, task, device Name, and face (application name) is added along with initializing timer. And Interest is forwarded using FIB. Above process is repeated until data is sent or Interest is sent to last node in the network.

Algorithm 2 describes the delivery of Data in response to received Interest. When any node finds corresponding Data to Interest, this node forwards this Data towards its neighboring nodes. And every neighboring node checks this Data Packet name exists in PIT to deliver the data. If it does not find corresponding entry, then this

Algorithm 2 Processing of Data in the Proposed Naming Scheme
Input: Data [Domain Name, Location, Task, Device Name, Content]
Output: Data[Receive Data at Sink Node]

```
1: Send Data to Neighboring Node(s).
2: repeat
3:     if Name in PIT then
       if Face has 0 Identifier then
          Sink Node Receives Data.
       else
          Cache Data in Intermediate Node(s).
          Forward Data to Next Node.
          Remove [Location, Task, Device Name, Face] from PIT
       end if
    else
       Drop Data.
    end if
4: until Data delivered to Sink Node
5: Remove Identifier 0 from Sink Node.
```

node drops the data (this node is not that who sent the request). If PIT entry exists then, it further checks that the packet has 0 identifier, the node has already received data. Otherwise, if this node is not consumer node then after caching the Data it forwards to next node and it removes this Interest relevant data from PIT. This whole procedure is repeated until data is delivered to sink (consumer) node. After receiving data, sink node stores and removes its identifier as 0.

4.3.4 Advantages of the Proposed Naming Scheme

We present some key advantages of the proposed naming scheme, which indicates its feasibility and viability for IoT environment and compared against related work in Table. 4.2:

4.3.4.1 High Aggregation

Name-aggregation is very useful for both reducing the number of entries in the routing table and efficient lookup operation. The hierarchical part of the proposed scheme provides great aggregation; this is the one reason to use it first. For example, consider two Interests having same prefix such as /uetTaxila/CP/DC/sensing:0x 9F091C54 and /uetTaxila/CP/DC/action/on:0x9F091C54. Because the prefix /uet-

Taxila/CP/DC/ is same for both Interests, so they will be aggregated in the routing table, which will minimize the number of entries in the routing table and reduce memory size of the router.

4.3.4.2 Fixed Length

The hierarchical names, which are composed of different parts can have variable length. In return, this affects the performance in terms of query speed in routing table or forwarding table. Therefore, the names should be of fixed length. The proposed scheme has been restricted to a maximum of 102 bytes for both Interest packet and Data packet. Out of these 102 bytes, last 4 bytes (32-bits) of Interest message and last 8 bytes (64-bits) of Data are reserved for flat part. Algorithms, which depend upon the length of query, can run efficiently if the length is fixed.

4.3.4.3 Scalability

The proposed naming scheme considers the concept of hierarchical naming which inherits high scalability (due to high name aggregation) as compared to the current IP-based communication systems.

4.3.4.4 Security

Proposed scheme tends to provide high level of security in form of integrity due to the use of flat component which is hashed (of device name and data message value) by FNV-1a hash algorithm.

4.3.4.5 Push Support

In IoT, when main application server needs to push any command towards data providers (i.e.,sensors) then proposed naming scheme's sub-component *Task* with the value of *action* provides push support.

In the following section, the proposed naming scheme is investigated for IoT-based application scenario.

4.4 Investigation of the Proposed Naming Scheme in IoT Domain

To investigate the proposed naming scheme in IoT domain, we consider Smart Campus (SC) as a reference model for IoT scenarios. A SC can include temperature sensors to exhibit important feature of temperature sensing. Temperature sensing feature can be used to control and monitor both labs and classrooms through the implementation of different actions like switching the air conditioners, computers and different sensors on and off. A CCN-based SC server (sink node) pulls (fetch) the data from sensors by sending Interest messages at regular intervals of time and then accordingly push (send) commands to perform above-mentioned actions.

There are different modes available for transmission of packets from source to the destination like unicast, multicast and broadcast.

4.4.1 Smart Campus Scenario Description

A hybrid model with both type of nodes (i.e., static and mobile) have been employed to analyze the efficiency in terms of sending and receiving of Interest and Data packets. Fig. 4.3 shows the simulation setup for the investigation of proposed naming scheme with different number of nodes as reference models. The devices used in scenario are sensors, routers and mobile data carriers for delivery of Interest messages to static nodes that are out of range. In reference model, there are eleven departments named Civil (CE), Mechanical (ME), Electrical (EE), Electronics (EC), Computer/Software (CP/SE), Transportation (TRE), Transport Office (TR), Library (LB) and Admin Block (AD), Telecom/Computer-Science Dept. (TE/CS), Industrial/Humanities Department (IE/HU). In each building, total five devices are placed at different locations for sensing temperature or switching on/off devices as mentioned in Interest packet. Out of these five devices, four are Zigbee-based Temperature sensors and one is Wifi Access Point (AP). Eight mobile data carriers are used which will travel on specific routes to forward and collect the requests and data respectively. A sink node (wifi) as SC server is placed in the CE because this is located at the center of all other nodes. Two mobile nodes and all the nodes in CE are placed in the range of sink node as shown in Fig. 4.3. All nodes except sink node work as a source of required content. We vary number of mobile nodes from two to eight and mobile nodes are used to

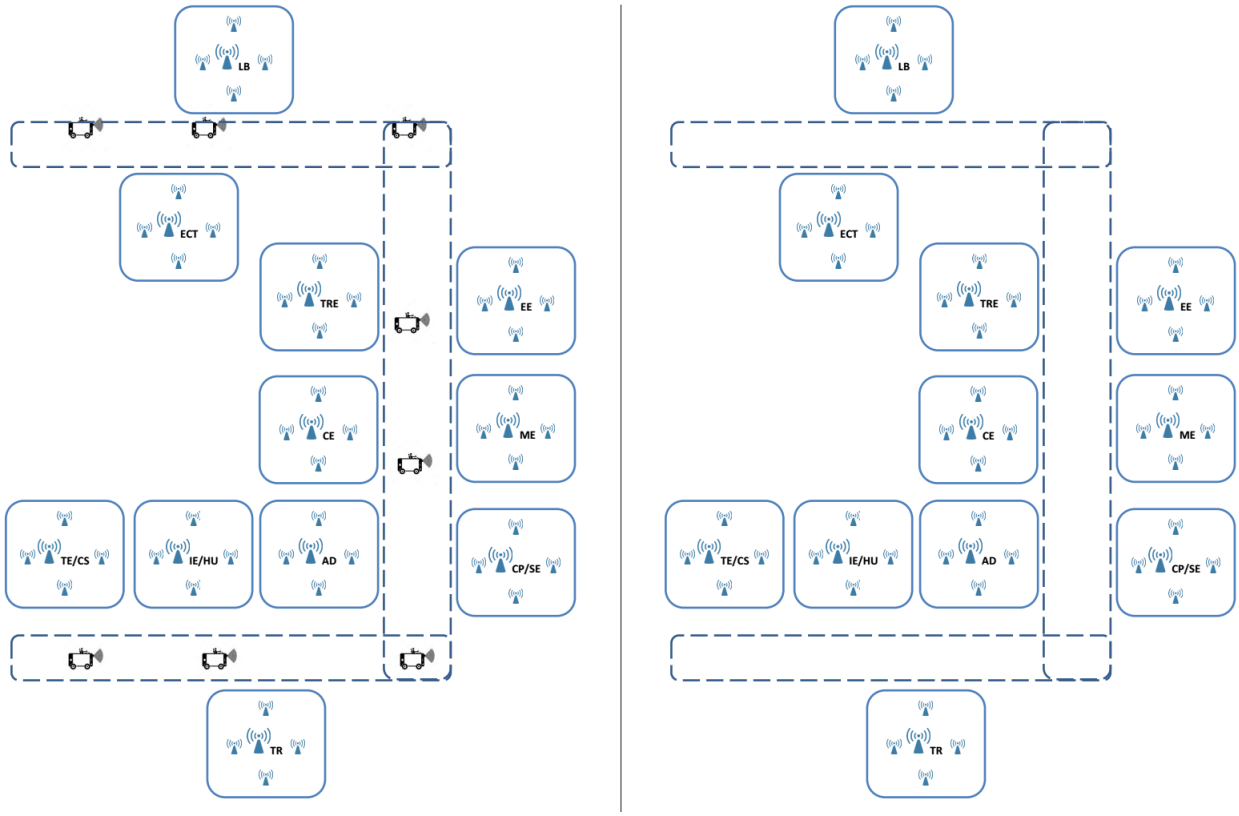

Figure 4.3: Reference Model for Proposed Naming Scheme, Left Side: with 8 Mobile Nodes and 55 Static Nodes and Right Side: Only Static Nodes, LB= Library, ECT= Electronics, EE= Electrical Eng, TRE= Transportation Engg., ME= Mechanical Eng., CE= Civil Eng., CP/SE= Computer/ Software., AD= Administration Block. TE/CS= Telecom/ Computer-Science, IE/HU= Industrial/ Humanities, TR= Transport Section.

forward Interest message towards static nodes deployed in other departments. In the shown scenario with eight mobile nodes, the sink node generates an Interest; the two mobile nodes receive it and start moving towards the other two (in both directions) that are in idle state.

Second scenario (Right side of Fig. 4.3) having only static nodes is used for comparison purposes. If a node cannot satisfy the Interest, it forwards to other nodes in its range. This chain is used to forward the Interest from the sink node to last source node in the given scenario.

4.4.2 Communication Loop Problem Solution

As CCN implements broadcast mode of transmission in order to send and receive Interest and Data packets respectively. This broadcast mode of communication presents a problem called 'loop problem'. When a sink node transmits an Interest to all nodes in the network range and stores an entry in its own PIT. If the receiving nodes cannot satisfy the incoming Interest, they store that Interest in their PIT and then forwards to other nodes. In that case, whether an Interest is satisfied or not, it remains in PIT of the sink node, which creates a loop. For example, if a sink node has two neighboring nodes and it broadcasts an Interest towards both of them. If the Interest is satisfied on the first node but not on the second. Then the first node sends CO towards both sink node and second node, while second forwards the Interest by broadcasting again towards the first node and remaining neighboring nodes. Both first node and sink node will receive same interest again. This situation creates a communication loop [136]. To resolve this problem, we use the concept of unicast using multi-hopping along with broadcast. For example, in the reference model, initially, sink node broadcasts an Interest message to all neighboring nodes, while multi-hop unicast is implemented for both to deliver the Interest packet to nodes in the department (on the basis of prefix mentioned in hierarchical component) and send Data packets back to the sink node.

4.4.3 Tree Structure of the Proposed Naming Scheme

We illustrate an example of proposed naming scheme (Details in Table. 4.3) through tree structure shown in Fig. 4.4. We can easily extend this by adding new locations and types of task. For example, an action can be performed on computers/air conditioners by sensing the temperatures of said location. An Interest message, e.g., /uetTaxila/CP/DC/action/on:C60B9850 is used to turn on the PC1 in DC Lab of Computer Department in UET, Taxila. After the Interest is satisfied, a Data packet will be sent back consisting of the name same as Interest and a 32-bit hashed value of a short string representing that the Interest has been satisfied or not. If the Interest is not satisfied, the payload of Data packet may have a specific hashed string to represent any malfunctioning of computer.

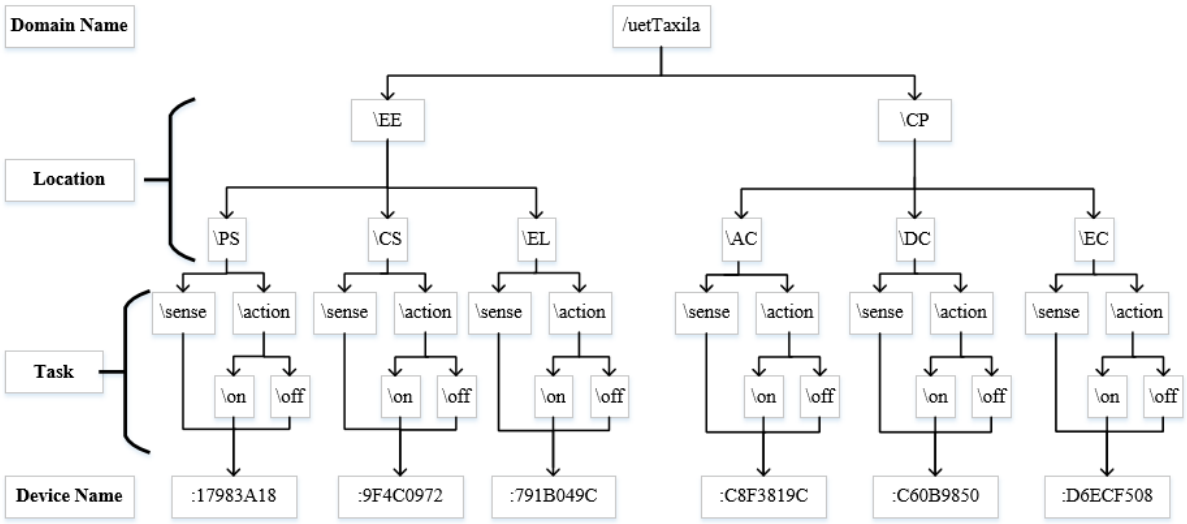

Figure 4.4: Tree Structure for Proposed Naming Scheme.

4.4.4 Processing of Interest and Data Message

Fig. 4.5 describes the exemplary scenario to highlight the working of the proposed scheme. For the sake of simplicity, it shows only the nodes that are involved in communication. In this example, the sink node generates an Interest message and broadcasts it to all the nodes in its range. MN1 receives the Interest message and stores it in PIT. If there is another entry with the same prefix, then it discards the Interest and adds a face identifier to the existing entry representing the original source of the Interest packet. Once the Interest is stored in PIT, node forwards it to other nodes in the connectivity. A mobile node is used for propagating the Interest in the network so that the nodes at a distance can receive the Interest message and perform operations accordingly. Fig. 4.5 shows different steps to understand the working of example scenario. Following are the steps of Interest generation and satisfaction process in the reference model:

1. As sink node is placed in CE department having one mobile node MN1 and all static nodes of CE department are in its range. Sink node generates an Interest with name /uetTaxila/EC/DC/action/on: C60B9850. It is stored in the PIT before propagation to other nodes connected to sink node. An identifier 0 is attached to the Interest stored in PIT of sink node indicating that it is the original requester of Data.

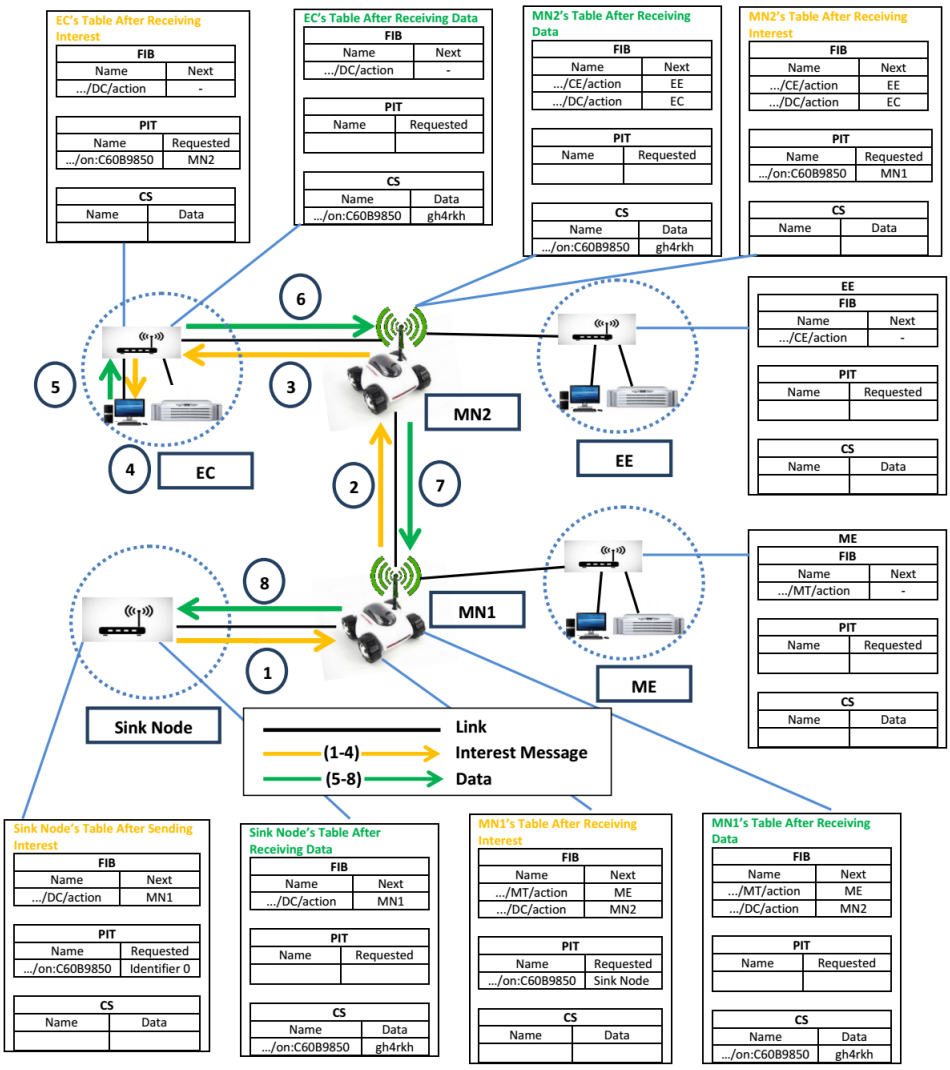

Figure 4.5: An Illustration of Proposed Naming Scheme in Reference Model.

2. Sink node transmits (broadcasts) generated Interest message to all the nodes in its range (MN1 and all static nodes of CE). As in this case, static nodes which resides in CE, discard this message by looking at the prefix. If the request is to retrieve data from nodes in other departments, mobile nodes help to forward the Interest message to nodes that are not in direct range of sink node. Therefore, MN1, after receiving Interest message from sink node, performs a long prefix match on FIB to check either this entry is already available or not. In the case of absence, MN1 starts moving on the specified route and forwards the Interest to all the faces stored in its FIB at regular intervals of time. But if the Interest is meant to perform a task in CE department where sink node resides, then

Interest is processed without the interference of any mobile node. CE static node receive Interest and send corresponding Data message after performing specified action in unicast mode.

3. Nodes at EE and ME do not process the Interest because their prefix doesn't match with the Interest prefix. The mobile node, MN1 forwards the Interest message to MN2 after entering in its range and start moving towards the EC, which is the destination location in this case.

4. Five static nodes in EC department receive the incoming Interest but only one can perform the action requested in the Interest message, i.e., turning on the computers in DC Lab in Electronic Department. It is because the nodes residing within the department implements the unicast mode of transmission.

5. After the Interest is satisfied, a Data message with Content Object (CO) is sent back to requesting node. The CO is /uetTaxila/EC/DC/action/on:C60B9850 | gh4rkh, where the prefix name of both the Interest message and CO message is same. The last hash/flat component of Data message contains the actual CO value.

6. PIT of EC removes the Interest from the list of unsatisfied Interests, stores the Data in its CS and then forwards to MN2.

7. CO message is replicated at intermediate nodes, MN2 and MN1, for fast retrieval of data in future.

8. The reference identifier 0 in PIT is removed, indicating that Interest has been satisfied. Finally, CO message is delivered to the sink node and stored in its CS indicating that action is performed successfully.

If there is a need to perform an action on a different number of nodes having same name prefix, then a prefix is sent as a name in the Interest messages instead of specifying a node. An Interest message with name /uetTaxila/EC/DC/action/on will turn on all the computers in DC Lab of EC department.

4.5 Performance Evaluation

In this section, we discuss performance evaluation of proposed naming scheme considering scenarios of smart campus shown in Fig. 4.3.

4.5.1 Simulation Environment

We use 55 static nodes and vary the number of mobile nodes (from two to eight) to cover the total area for exchange of Interest and Data messages among sink node and source nodes. All nodes are arranged in the form of considered scenario. CCNx implemenations are used in Contiki OS based Cooja simulator [136]. We implemented proposed naming scheme in CCNx implementations in Contiki OS based Cooja simulator installed in a Linux Ubuntu running on Virtual Machine. A core-i5 PC with 4GB RAM is used for implementation and performance evaluation. CCNx is modified in accordance with the proposed naming scheme. CCNx hierarchical naming is concatenated with flat naming by implementing a separate function to calculate hash (pseudo code is given above). Simulations are run for 1600s and Tmote Sky type motes are used. We use Random-Waypoint Mobility Model (RWMM) mobility plugin in this simulation for mobile nodes to move in particular direction. RWMM has the following input parameters: node number, time in seconds, x-coordinate, y-coordinate. The node number is not the Rime address and it starts from '0' means the 'node 0' represents the 'mote 1'. The coordinates have value in meters. Rest of the simulation parameters that help in evaluating the proposed technique through a series of simulations in Contiki-based Cooja Simulator are summarized in Table 4.4. Though fixed number of nodes are considered in these scenarios but this implementation can also be validated on a large scale as the proposed scheme presents high scalability.

To compare and show the worth of proposed naming scheme, three different scenarios (according to Fig. 4.3) are simulated in Cooja Simulator. Fig. 4.6 shows the simulation of only static nodes in which intermediate nodes forwards the Interest packets. The simulation of delivery of Interest message and Data message using a different number of mobile nodes is shown in Fig. 4.7 and Fig. 4.8. Each Figure contains three parts. Part 1 shows the start of simulation when the sink node sends the Interest to its neighboring nodes. If the nodes cannot satisfy the Interest, it is

Table 4.4: Simulation Parameters

Parameter	**Value/Name**
Communication Stack	CCN
Radio Medium	Unit Disk Graph Medium (UDGM): Distance Loss
Topology Size	200*350
Number of Nodes	55 Static, 8 Mobile
Number of Transmissions	200
Tx Range	40 meters
Interference Range	60 meters
Packet Size	102 bytes
MAC Layer	IEEE 802.15.4
Network Driver	ccn_driver
MAC Driver	csma_driver
Radio Duty Cycling Driver	sicslowmac_driver
Radio Driver	cc2420_driver
Mote Type	Tmote Sky
Speed	No limit speed
Simulation Time	1600 s
Mobility Model	Random-Waypoint Mobility Model (RWMM) [163]

forwarded to other nodes using mobile nodes (part 2). Part 3 shows the case when the Interest is forwarded to the last nodes in the campus area and Interest forwarding is finished in this step.

4.5.2 Performance Metrics

The performance of proposed scheme is evaluated through following metrics:

- Satisfaction Rate (SR): the ratio of total number of satisfied Interests to total number of generated Interests and can be calculated by using equation 4.1 as follows:

$$SR = \frac{\sum_1^N SatisfiedInterests}{\sum_1^n GeneratedInterest} * 100 \tag{4.1}$$

Where 'n' is the total number of generated Interests and 'N' is the total number of satisfied Interests.

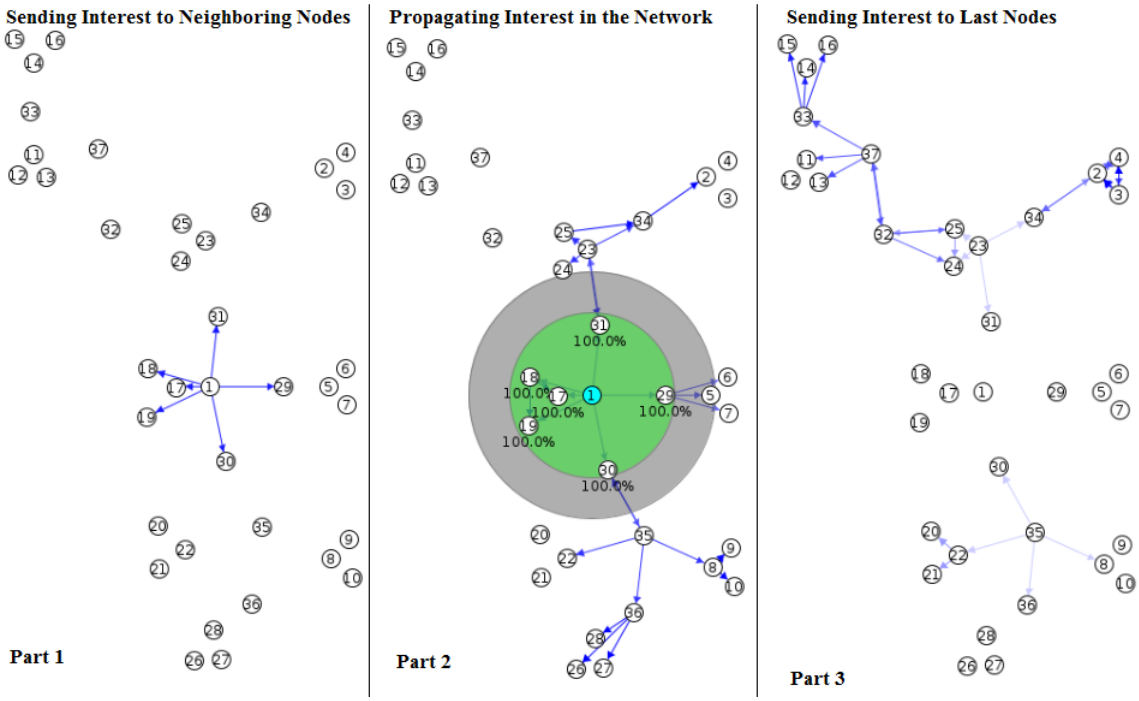

Figure 4.6: Simulation Scenario 1 (Only Static Nodes).

- Average Latency: is the ratio of total latency to the total number of packets received at sink node and can be calculated using following equation 4.2:

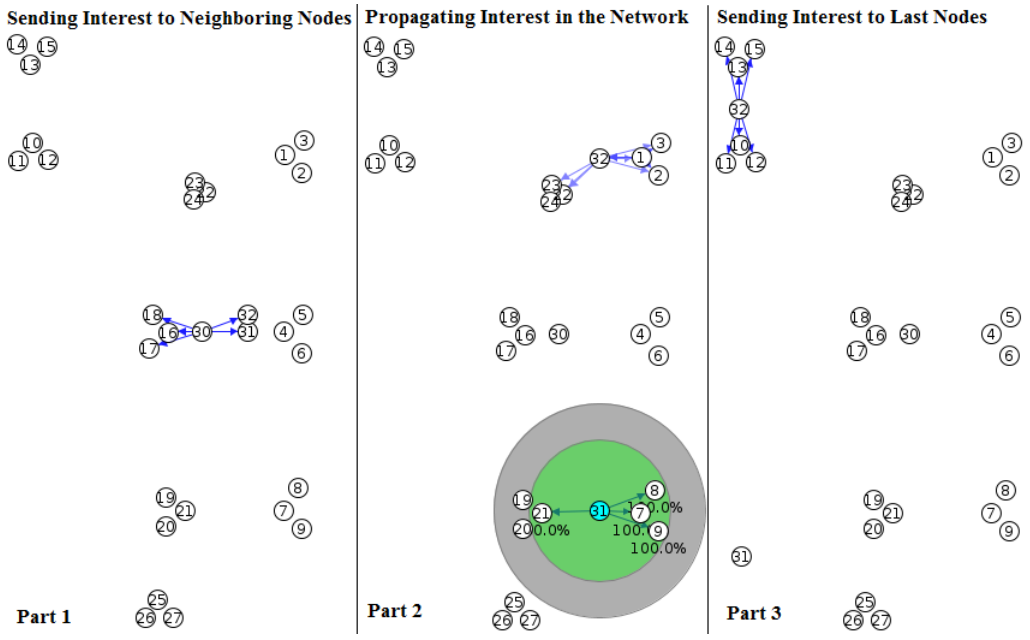

Figure 4.7: Simulation Scenario 2 (2 Mobile Nodes).

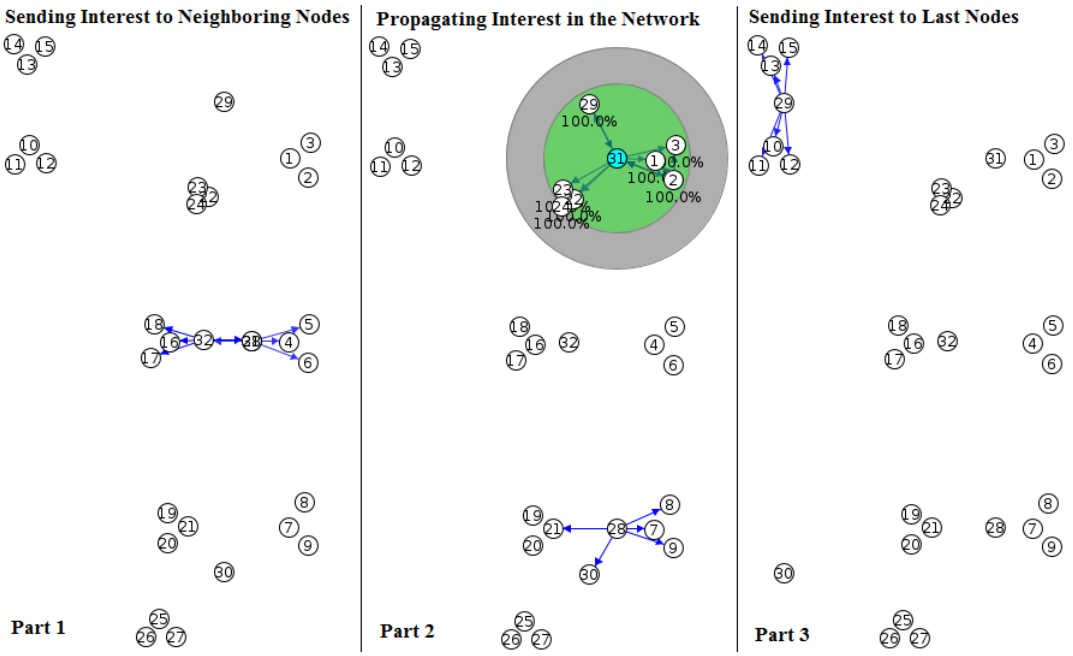

Figure 4.8: Simulation Scenario 3 (4 Mobile Nodes)

$$AverageLatency = \frac{(TotalLatency)}{(TotalPacketsReceived)} \tag{4.2}$$

where total latency is the delay between Interest generation by sink node and its reception at the source node and can be calculated using following equation 4.3:

$$TotalLatency = \sum_{n=1}^{n}(RecvTime - SentTime) \tag{4.3}$$

The aggregation, number of hops and Interest transmissions are computed using the log files generated for the simulations performed.

4.5.3 Results and Discussion

4.5.3.1 Satisfaction Rate (SR)

Because static nodes implement only 'broadcast mode', and broadcast mode is the major reason of 'communication loop' creation. Therefore, due to presence of 'communication loop', low SR is an obvious result and it is evident from Fig. 4.9 that static nodes exhibit low Interest SR.

As we stated above, we minimize 'communication loop' in scenarios which have mobile nodes by implementing 'unicast mode' in source nodes that reside in depart-

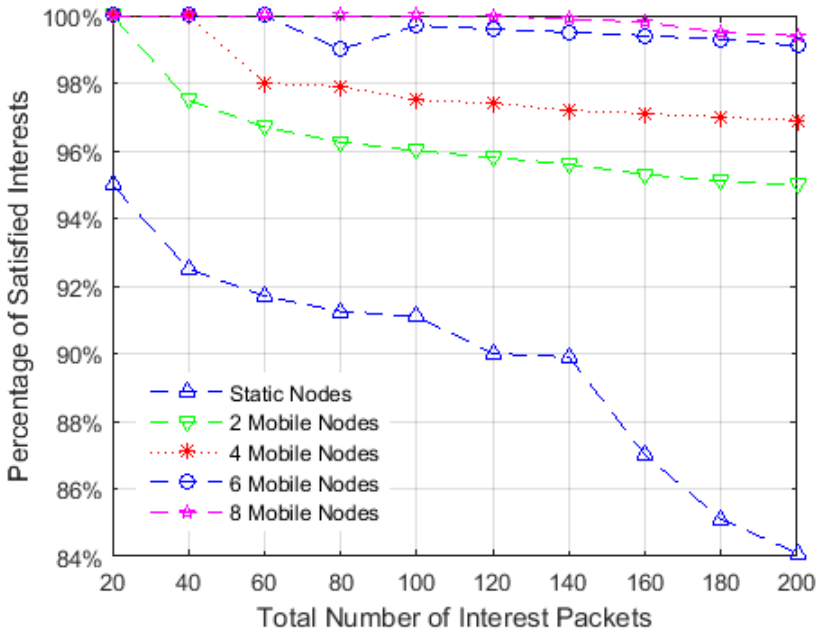

Figure 4.9: Percentage of Satisfied Interests.

ments (i.e., source nodes send data to the requesting mobile node only instead of sending to all neighboring nodes). In this way, the percentage of the average number of Interests satisfied, increased. This results in higher SR. For simulation, a number of Interests are generated and their satisfaction rate is over served by receiving the Data requested. Mobile nodes shows better results because more average number of Interests are satisfied before forwarding it to next mobile nodes and it is satisfied before its expiry in the given time period.

4.5.3.2 Average Latency

Average amount of delay in receiving the Interests is higher in mobile nodes scenarios. Reason of this delay is the time which mobile nodes take while traveling from sink node to destination node. Although static nodes show less delay but they have more number of intermediate nodes between the sink and source nodes. Fig. 4.10 shows average latency computed for different number of Interests in considered scenarios. The average delay in the scenarios with more than two nodes is not much higher because the sink node has some nodes in its range and some on very short distance. With the help of log generated for simulations, it is observed that more Interests are satisfied before forwarding the Interest towards next mobile nodes which result less

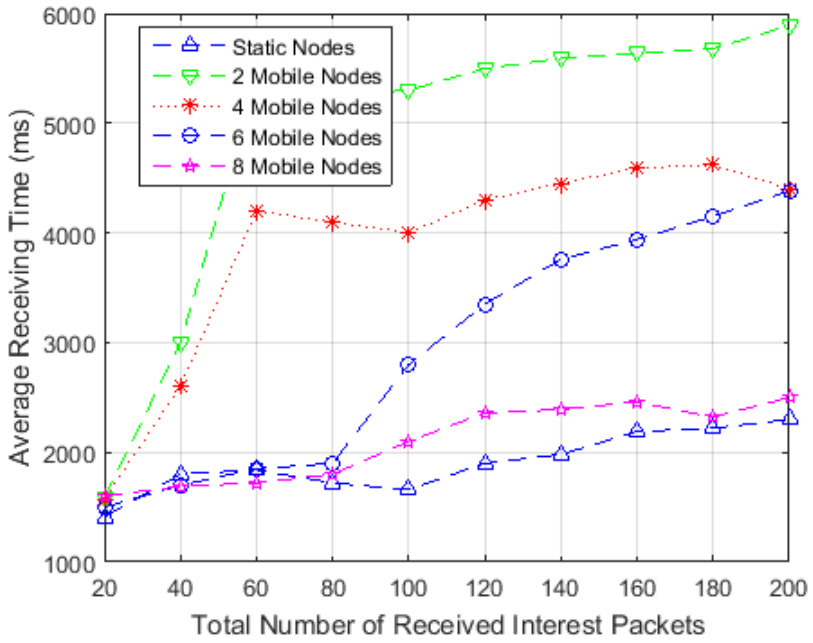

Figure 4.10: Average Delay of Receiving Interest.

average delay in comparison to two mobile nodes which move back after completing the path.

4.5.3.3 Transmission of Interest Packets

The average number of transmissions are higher for static nodes because of the greater number of intermediate nodes and broadcasting mode. Every node, which receives the Interest packet, forwards the Interest to neighboring nodes if it does not have the required Data. The mobile nodes work as carrier nodes to take the Interest packets to nodes at a distance, minimizing the number of hops. They require less number of transmissions because only the mobile nodes works as intermediate nodes. Fig. 4.11 shows the comparison of transmission of Interest packets to reach the source node.

4.5.3.4 Number of Hops

Fig. 4.12 highlights and compares the performance of the proposed scheme in terms of number of Interests across different hops. Different number of mobile nodes are presented for detailed investigation. In the case of static nodes, transmission of Interest and Data packets is done with the help of intermediate static nodes only. The sink node broadcasts the Interest to its neighboring nodes. If the Interest is not satisfied,

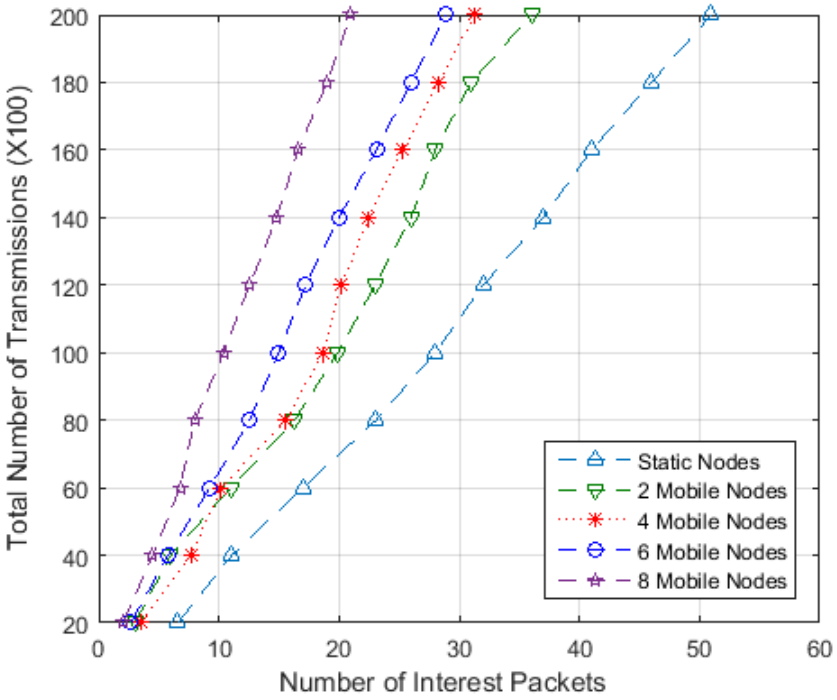

Figure 4.11: Total Number of Transmissions (One Sink Node).

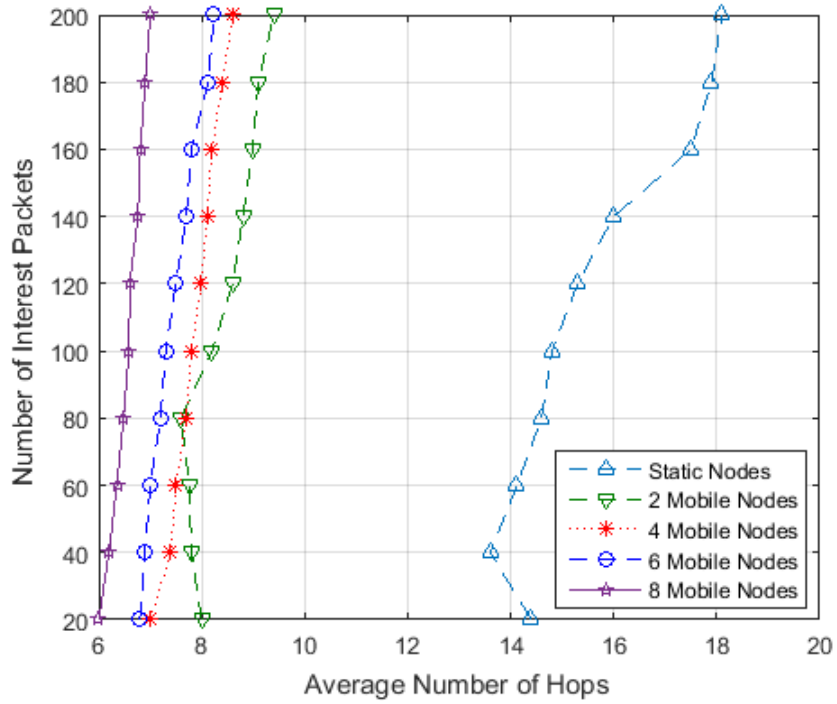

Figure 4.12: Average Number of Hops.

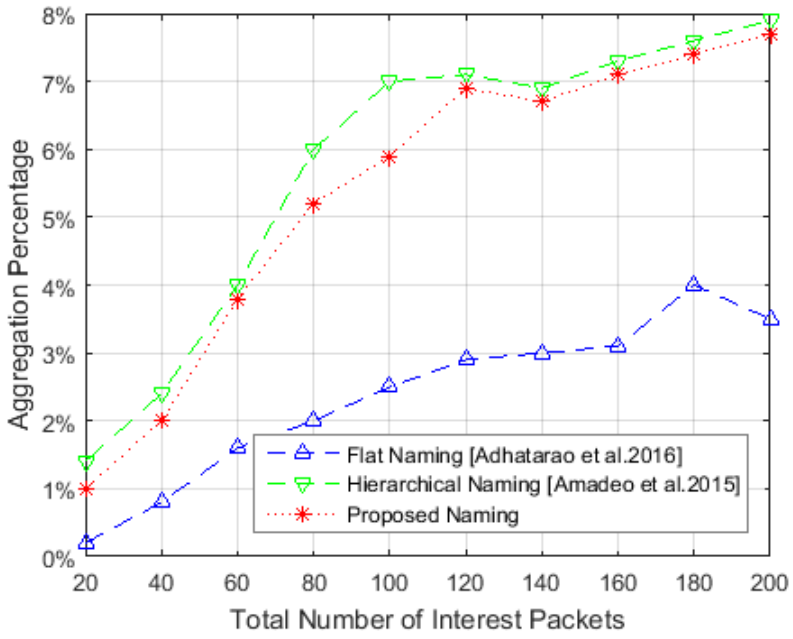

Figure 4.13: Percentage of Interest Aggregation.

then it is propagated to next nodes until it is satisfied and Data packet is received. For this reason, it has a greater number of hops in comparison to those where mobile nodes transmit Interest and Data packets between the sink and last source node.

4.5.3.5 Interest Aggregation

The aggregation of Interests in the intermediate nodes minimizes the size of routing table (PIT and FIB) and number of entries in it. Fig. 4.13 shows the percentage of Interests aggregated in considered scenarios. It can be seen that flat naming shows very low aggregation as compared to hierarchical naming. Since, the proposed hybrid-naming scheme has both flat and hierarchical naming components, therefore it has higher Interest aggregation in comparison to only the flat naming scheme but less as compared to hierarchical naming. The Interests with similar prefix require only one entry in the routing table and it is differentiated by the flat part, which is unique to every component. The results are obtained by observing that how many Interests are generated and stored in the routing table on average.

4.6 Chapter Summary

In this chapter, we presented and simulated a CCN model for IoT-based SC application, in which a hybrid naming scheme is proposed where contents are named through the use of both (1) hierarchical and (2) flat naming schemes. The hierarchical component takes the domain name, location, task and device name in URL style. By using 'task', push support is added in the native CCN protocol. The flat component is used to provide integrity and it is computed through the FNV-1a hash of the device name and Data. The communication loop problem associated with CCN protocol is eliminated by implementing 'unicast' protocol on the source nodes. Mobile IoT nodes are used for delivery of Interest and Data packets to nodes that are not in the range of sink node. The proposed naming scheme is evaluated for IoT-SC having both static and mobile nodes and results revealed the significant gain in terms of success rate and number of transmissions of Interest packets, latency, number of hops and interest aggregation.

Chapter 5

Extended NDN-IoT Naming and Forwarding Scheme: Holistic Naming and Basic Forwarding for NDN-based Internet of Things: The Case of Smart Campus

Information-Centric Networking (ICN) specifically Name Data Networking (NDN) is the name-base (content-base) networking and takes named-contents as "first class citizen", being considered as the ideal candidate to form the Future Internet basis. NDN striking features like named-data self-secured contents, name-base-forwarding, in-network caching and mobility support suits the Internet of Things (IoT) environment, which aims to enable communication among smart devices and to combine all Internet-based smart applications under the one roof. With these aims, IoT put many research challenges regarding its network architecture as it should support heterogeneous devices and offer scalability. IoT may depend on the names and addresses of billions of the devices and should smartly manage the bulk of data produced every second. IoT application smart campus has gained a lot of attention in both industry and academia due to many reasons. Therefore, to design NDN for IoT, a sophisticated naming scheme is needed to explore and it is the main motivation for this work. In this chapter, we study NDN-IoT smart campus (in terms of connected devices and contents) and find that it lacks in a reasonable naming and addressing mechanism; and thus we propose NDN-based Hybrid Naming Scheme (NDN-HNS) for IoT-based Smart Campus (IoTSC).

Internet of Things (IoT) is on the journey to combine all things or devices under one umbrella to enable global access to information and connectivity among Internet-based devices. The connectivity provided and information produced by these huge number of devices can be easily handled by future Internet architecture namely Information-Centric Networking (ICN). There have been myriad future Internet architectures that use the basis of ICN, and among them Named Data Networking (NDN) is a promising one. Incorporating NDN with IoT-based application brings solutions to many issues however this fusion makes accessing the IoT content more easily provided that a sophisticated naming scheme is designed to perform this task. In this chapter, we design a NDN-based hybrid naming scheme (NDN-HNS) and implement this naming scheme on our own designed secure forwarding schemes (NDN-HNFS) for consumer, producer and content router. We consider IoT-based smart campus (IoTSC) scenario for design and evaluation which is a true representative scenario of IoT due to it's scalability, heterogeneity, and security requirements. On the basis of NDN-HNS, we provide all activities list divided into two communication models push-based model (PHTC) and pull-based model (PLTC) that can be generalized to any IoT application. We evaluate the NDN-HNFS against legacy NDN in terms of interest satisfaction rate, delay, and number of transmissions. The results prove that NDN-HNFS perform better and in an optimal way than legacy NDN.

5.1 Introduction

As smart connected devices usage prevalent now-a-days. And plethora of these smart connected devices is referred as Internet of Things (IoT) [164]-[165]-[166]. These smart connected devices may include smart phones, tablets, smart gadgets like iWatch, smart glass, smart brush, smart AC and gained much attention from consumers as well as from both industry and research societies. For example, IoT European Research Cluster (IERC) aims to identify the challenges to transform the IoT concept into IoT reality. IERC involves other countries than Europe to address global connectivity [1]. Another, ICN Research Group (ICNRG) is also trying to gather requirements to build IoT architecture using ICN [156]. But, with the advent of IoT, its efficient network architecture is still in its infancy.

In the context of ICN, architectures are proposed like NDN, PURSUIT, SAIL, MobilityFirst, CONVERGENCE, COMET, Green ICN and C-DAX [35]-[127]. ICN has many research areas to be explored yet and naming the contents is one of them. Among these ICN architectures, NDN is implemented as proof-of-concept and gaining a lot of attraction through research community [33].

Hierarchical naming scheme is used to name the contents in NDN. This naming scheme provides long variable length names [35].

On the other hand, in general, IoTs contents are ephemeral, short-lived, small, fresh, different priorities and from different locations. Therefore IoTs contents have special features and factors that need to be accommodated/ considered while naming any specific content.

In this chapter, we select ***"Smart Campus"*** as an IoT usecase, as shown in Fig. 5.1. We take smart campus because it involves almost all operations that any other IoT application does and research for smart campus is a never ending process as we can keep adding smart services to make it more smarter. At the same time, it is very emerging research topic from both academics and industry perspectives[167]-[168]. Moreover, in cases like smart home, IoTs scalibility may not be estimated appropriately due to limited number of nodes and data.

Further, we state IoT-based Smart Campus (IoTSC) as a building or combination of many buildings that may utilize(s) many smart sensors (temperature, humidity, pressure, proximity and etc.,), actuators (AC, lights, fans, doors, windows, vehicles, mobile phones, alarm buzzers) and connecting devices (Ethernet, Wifi, Bluetooth) to provide anytime connectivity. CAmpus Server (CAS) is responsible to provide connectivity among all devices and monitor all activities as controlling agent as can be seen in Fig. 5.1. NDN-based IoTSC application can provide many services as energy management, security and privacy. Moreover, Information-Centric Networking Research Group (ICNRG) [24] which is an IETF-based project has started to explore ICN in terms of IoT [156]. Although there are many areas yet to be explored for IoT as highlighted in [127]-[24] and content naming is one of these. We argue that to build IoT on the basis of ICN, naming is foremost issue which should be resolved primarily. Because other issues like security, in-network caching, can be build on sophisticated naming scheme. Information-Centric Networking in this

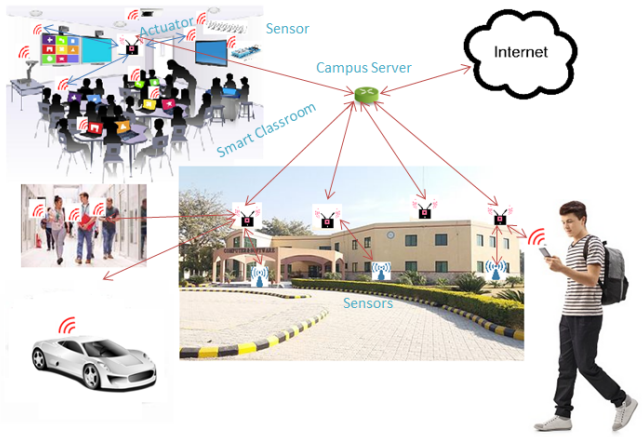

Figure 5.1: Smart Campus, Computer Engineering Department (CPED), UET, Taxila.

perspective provides three naming mechanisms [132]:, (i) hierarchical naming, (ii) flat-based naming and (iii) attribute-based naming. Hierarchical naming is used by CCN and NDN while self-certifying flat-based naming is used by DONA, COMET, SAIL, PURSUIT, CONVERGENCE and MobilityFirst. Moreover, attribute-based naming which is followed by CBCB can be used along with other two schemes, thus resulting in hybrid naming scheme [132]. Every naming scheme has its own advantages and disadvantages according to its usage [132]. However hybrid naming scheme can play an important role and thus we explored it while combining hierarchical and flat naming schemes in our previous work [141]. The real power of ICN in IoT lies in the naming of information (content). IoT contents have some notable properties, for example, they are ephemeral, oftenly require location information, different freshness and probability. Thus, in this chapter, to support attribute-based naming (which can be very useful in lookup with attributes), we combine hierarchical, flat-based self-certifying to form NDN-based hybrid naming scheme (NDN-HNS) to name contents and devices considering best features from ICN naming schemes [131] for IoTs environment. Providing holistic naming scheme for NDN-IoT is the major contribution of this work. In this scheme, name is assigned to content in three components: IoT application prefix, hierarchical, flat-hash and attribute-based. Proposed naming scheme NDN-HNS can provide MTC support, scalability, security and addressing and naming to data contents and devices. Then on the basis of proposed naming scheme NDN-HNS, we categorized IoT traffic types into seven types which we used to design

ICN-IoT to enable machine type communication (MTC). MTC is the main pillar of IoT and we designed NDN-IoT in a way to support both push and pull-based communication models. This is done by our NDN-HNS-based forwarding schemes which we call NDN-based Hybrid Naming and Forwarding Schemes (NDN-HNFS). NDN-HNFS are designated due to the fact that these forwarding schemes for NDN producer, consumer and content routers(CRs) are designed on the basis of NDN-HNS. Besides these, light version of OnboardICNg, an authentication and authorization scheme, is used to ensure secure forwarding. The summary of our main contributions are as follows:

1. We propose NDN-based hybrid naming scheme (NDN-HNS) which incorporate content and device properties, and names through three components, hierarchical component (HC), attribute component (AC) and flat component (FC).

2. To achieve MTC, we incorporate content name parts as *task and sub-task* to decide message type and thus communication type as pull or push. NDN-IoT producer is enabled to send data (either critical or subscribed update) towards consumer by its own without request from consumer. This is how, we enable push type communication (PHTC).

3. To maintain content security in terms of integrity, we include hashes of content name, sub-name and IDs of devices at the end of the message. We also use this hashed information in OnboardICNg to authenticate and authorize the message sender to validate about its right to perform action.

4. We propose forwarding schemes NDN-HNFS for consumer, producer and content router. These NDN-HNFS ensure that the message is received/sent by tentative node.

5. We simulate IoTSC with both static and mobile nodes in ndnSIM and evaluate naming and forwarding schemes. We find that our proposed schemes outperform legacy NDN in terms of number of transmissions and delay to perform pull- and push-based activities of IoTSC.

Rest of the chapter follows with importance of smart campus w.r.t IoT and NDN-based research efforts with some background of how TCP/IP is less suitable for IoT,

NDN basics and naming mechanisms and NDN naming schemes for IoT in Section 5.2. Updated IoT application categorization, proposed NDN-HNS and its important parts are described in Section 5.3. In Section 5.4, we describe proposed working of proposed naming scheme NDN-HNS with the help of two communication models and following forwarding scheme NDN-HNFS. Simulation and implementation details along with results and discussions are discussed in Section 5.5 and finally we conclude and summarize our chapter in Section 5.6.

5.2 IoT-based Smart Campus and NDN Related Research Efforts

This section firstly, presents an overview of smart campus importance in IoT from both academics and industrial perspectives. Further we describe why TCP/IP is less suitable for IoT. Then we discuss some research efforts in which NDN is considered for IoT and smart campus.

Google funded and started GIoTTO program at CMU with the collaborators from other universities as well to transform IoT into reality and under GIoTTO they build lab to convert CMU into smart campus[167]. They designed GIoTTO open source stack to provide support for heterogeneous platforms. Moreover, Google invested in NEST Labs for IoT, which aim to build smart home [168]. On the other hand, Huawei used IPv4 and IPv6 to provide smart campus services to Huawei Hubei University of Technology [169]. They aim to provide smart agile campus, cloud data center, disaster management and recovery mode, easy access to library and safety of campus through video surveillance. In addition, Pakistani government has announced that thirteen universities are being converted to smart campus by the end of year 2017 [170]. As a first step, Pakistani government has decided to provide students and faculty with free wifi everywhere inside the campus. Next step is to utilize campus resources to monitor energy usage to control and optimize its consumption.

Major activities/services that can help to build IoTSC may involve optimization and control of energy usage [171], security of faculty and students vehicles, security and privacy of data of both faculty and students, easy access of library resources, smart cafeteria ordering and students behavior analysis. Other advance services may include disaster management, hostel allotment and mess management system, smart

attendance system and smart time table management system. Mostly services can be achieved through simple operation of connectivity among smart devices i.e., both sensors and actuators. While other services may be achieved through smart mobile applications development, cloud computing, artificial intelligence algorithms and neural networks [172]-[173].

5.2.1 Why TCP/IP is Less Suitable for IoT

In this subsection, we argue that employing TCP/IP architecture for IoTSC can burdensome constraint-oriented devices in many ways, for example:

i) Data produced by billions tiny IoT devices is huge in amount and needs better management to use this data for analytics applications,

ii) Mostly IoT devices (especially wireless sensors) come up in small memory due to the processor used and the circuit board design. Small memory makes data unavailable frequently, further, there is no caching system in IP-based solutions which also raises data unavailability issue,

iii) IoT can definitely have devices that differ in specifications (differ in memory, power, communication range and processing power) and connection technologies (UWB, RFID, Zigbee, Ethernet, Bluetooth, Wi-Fi, Wi-MAX, MANETs, Cellular Networks, Radars and Satellite Networks (Although last two technologies are not meant for smart campus but may involve in IoT))[174],

iv) Scanning and inspection of transmitted data need complex and costly methods like Deep Packet Inspection (DPI),

v) Separate patches like IP-Sec protocols are employed to provide appropriate security and

vi) TCP/IP provides mobility support in complex way where multiple additional registrations (during mobile devices hand-off) are required to support mobile devices. Security requirements and mobile devices further complicates its applicability for IoTs.

Therefore, heavy TCP/IP architecture, either in native or in overlay manner, is less suitable for IoTSC.

In the meantime, many research efforts suggest ICN, specifically NDN for IoT smart environment.

5.2.2 NDN Basics and Naming Mechanism

NDN is one of the most promising projects of ICN. One of the reasons of NDN being such project is active research going on it [35]. NDN works on publish/subscribe model and offer name base networking through two messages namely Interest Message and Data Message, and three data structures namely Pending Interest Table (PIT), Forwarding Information Base (FIB) and Content Store (CS) [157]. Each of these data structures have different roles and priorities. Whenever an interest against a content object is received, it is first matched in the CS. If content is found, it is replied back. In case the content object is not in CS, an entry for the received interest is created in PIT. After that the interest is forwarded on the interfaces or faces using FIB. So the CS has the top priority following PIT and at the last comes FIB. Furthermore, the legacy NDN supports only pull mechanism. In which a consumer requests for a content object by issuing an interest to the producer of the content object. However, there are some applications that require pull-based mechanism overridden with push-based mechanism and IoT-based applications are some of them.

5.2.3 Related NDN based Naming Schemes for IoT

NDN has been implemented for smart home in both [36]-[175]. Both of these works discuss naive NDN hierarchical naming scheme for smart home. Further [176] implement NDN for lightning the home and found NDN more useful when traffic is local. They also implement NDN basic naming scheme for naming the data. Smart campus lightning system is implemented through NDN in [177]. They also use basic NDN hierarchical naming scheme to name the contents. NDN is implemented for under-water networks in [159] to build smart water. They also designed and implemented hierarchical naming scheme to name the water related contents. ICN based naming scheme along with encoding scheme is designed for IoT in [178]. Summary of most related works is presented in Table 5.1.

Table 5.1: Summary of ICN-based Related Naming Schemes

Application	**Naming Scheme Format (Example)**	**Traffic Model Support**		**Security**	**Device Info**	**Name Length**
		Push	**Pull**			
ICN-based naming for Smart Home [36]	/homeID/task/ type/subtype/ location/	✓	✓	✗	✗	long
CCN-based Naming for VANETS [114]	VHN://CTRY/STT/ CTY/OID/VRP /TYPE/Sub-Type/ SP/TEM/Attrs ./hash(Base64)	✗	✓	✓	✓	very long
NDN-based Naming for smart building [122]	/ndn/ucla.edu /bms/building/ melnitz/studio/1/data /panel/J/voltage/	✗	✓	✗	✗	very long
NDN-based Naming for under water [159]	UNDN://spatial/ (22.228,288.30,96) /temporal /2622487800 2682489000/type /Salinity/pref/all	✗	✓	✗	✗	medium
ICN-based naming for IoT [178]	/Bit:edu/ CentralBuilding /Floor10/Room33	✗	✓	✗	✗	long
ICN-based naming for IoT [141]	/domain-name /location/task: device-name| data /uettaxila/CP/ DC/action:on /0x9F091C54	✓	✓	✓	✓	long

These schemes name the content through only hierarchical naming mechanism without content attributes, push support and security. However in IoT, MTC is very essential which require both pull and push type communications along with important perspective of security. To support all of these essentials in a much better way, we propose hybrid naming scheme NDN-HNS to name IoTSC contents in a meaningful way. NDN-HNS also support security in terms of content integrity, and authentication

and authorization of message sender to validate about tentative device. Moreover, forwarding strategies help to receive data to only intended devices.

5.3 Hybrid ICN-based Naming Scheme for IoTs

This section is presented with threefold purposes: in first subsection we present updated IoT application categorization and in second subsection hybrid naming concept is described. In the third subsection, naming scheme components are discussed.

5.3.1 IoT Application Categorization

We use [179] to categorize IoT applications and show in Fig. 5.2. They listed twelve major categories and we updated (and somehow modified) this list by adding two more categories named as smart education learning and smart buildings.

5.3.1.1 Smart Cities

This application can include structure health monitoring (SHM), roads and their lighting management, traffic congestion management, parking lots management, waste management and noise control to build smart city.

5.3.1.2 Smart Water

This scenario can incorporates water monitoring through pollution levels in the seas and rivers, water level monitoring for flood control and chemicals proportion monitoring for purity water.

5.3.1.3 Smart Grid

This application can handle smart metering, electricity usage control and automatic billing.

5.3.1.4 Smart Environment

This can incorporate earthquake detection, snow level and landslide monitoring and management, fire detection and control in the towns and in the forests to make environment safe and sound to provide better life.

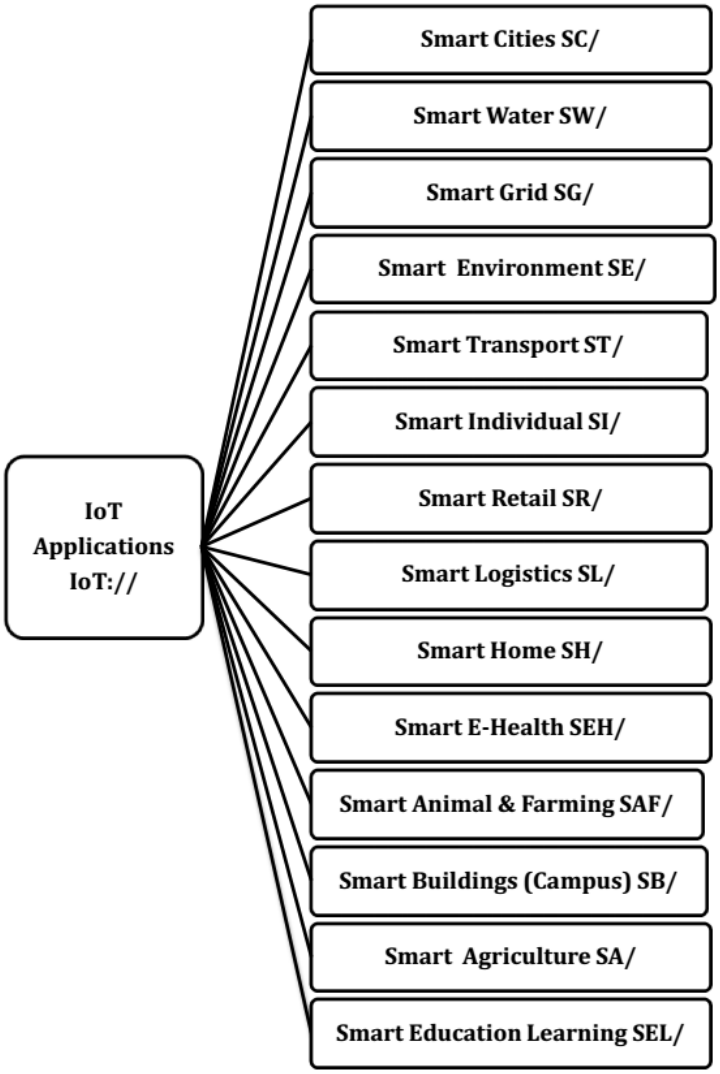

Figure 5.2: IoT Applications Categorization.

5.3.1.5 Smart Transportation

Efficient transfer of wide range of commodities can be included in this scenario. Vehicles carrying a commodity can be tagged and located via GPRS software.

5.3.1.6 Smart Individual

As mentioned in [180] IoT through smart wearable can be simple IoMe, to control your daily intake of calories, to share health measurements with your doctor and to motivate you to be more active in sports.

5.3.1.7 Smart Buildings

In [181], smart building can incorporate chillers, air handlers, automatic parking, energy consumption control and temperature and humidity monitoring.

5.3.1.8 Smart Logistics

This can incorporate planning, policy and infrastructure to manage route control, easy detection of an item in supply chain, quality control of shipments to ensure safety and storage compatibility management.

5.3.1.9 Smart Home

Security system to detect and monitor entrance of non-authorize persons, automatic control (locking and opening) of house doors and windows, consumption control of water, gas and electricity, remote control and monitor of house appliances can be combined to build smart home.

5.3.1.10 Smart Retail

This can include applications like smart shopping and payment management, product store management, payment management for parks, roads and gyms.

5.3.1.11 Smart E-Health

Check and control of sugar level and blood pressure, measurement of daily calories intake, assistance of elderly Alzheimer patients and proper medication (in case of emergency) through remote help from doctors can incorporate in providing smart-e-health.

5.3.1.12 Smart Animal & Farming

Quality control and improvement in survival rate of offspring, betterment in ventilation conditions, animal tracking and control of required conditions to ensure high quality crops can be combined to form smart animal farming.

5.3.1.13 Smart Agriculture

This can monitor and control soil moisture level, humidity, temperature and wind changes to maximize product (fruits and vegetables) quality and quantity.

5.3.1.14 Smart Education Learning

This scenario can include smart campus services like notification for exam results, evaluation of assignments and exams, and important announcements (about class timings and venues). Moreover, it can include energy management in campus.

5.3.2 Proposed NDN-HNS for IoTSC

When an IoT application user (for example IoTSC user) requests for a content, NDN Interest message is forwarded from the requesting node to nearest node (or intermediate node). This node further forwards Interest message to the node containing the content. Any node that originates data or caches data can reply DATA message. NDN model follows hierarchical naming scheme to name IoT contents. Hierarchical names are long and describe detail of content separating through /. NDN naming scheme is not designed for IoT applications as these application put many new constraints to name the contents. To fulfill IoT application specific requirements we propose a hybrid and holistic naming scheme which contains multiple parts to name any content. These parts or components are separated through the symbol ':', that indicates the end of current part and start of the next part. Moreover, :/ is used to specify multiple attributes of the content in attributes component and multiple sub-parts of flat component. While portions of these parts are specified through /.

5.3.3 NDN-HNS Components

Our hybrid naming approach contains following three parts. These parts are aggregated with the aim to provide scalable, secure and easy-to-manipulate naming scheme. But before these name parts begin, user has to specify primary root prefix as *IoT://SBC* for *Smart Building (Campus) IoT application* as it can be seen in Fig. 5.3. Then name follows with hierarchical component as secondary hierarchical root prefix, attribute component and flat self-certifying component.

5.3.3.1 Hierarchical Component (HC) or Secondary Root Prefix (SRP)

This part names a content in the same way as NDN assigns name. Hierarchical component (HC) combines campus information, content originating node ID along with content information. However, campus location information has nothing to do

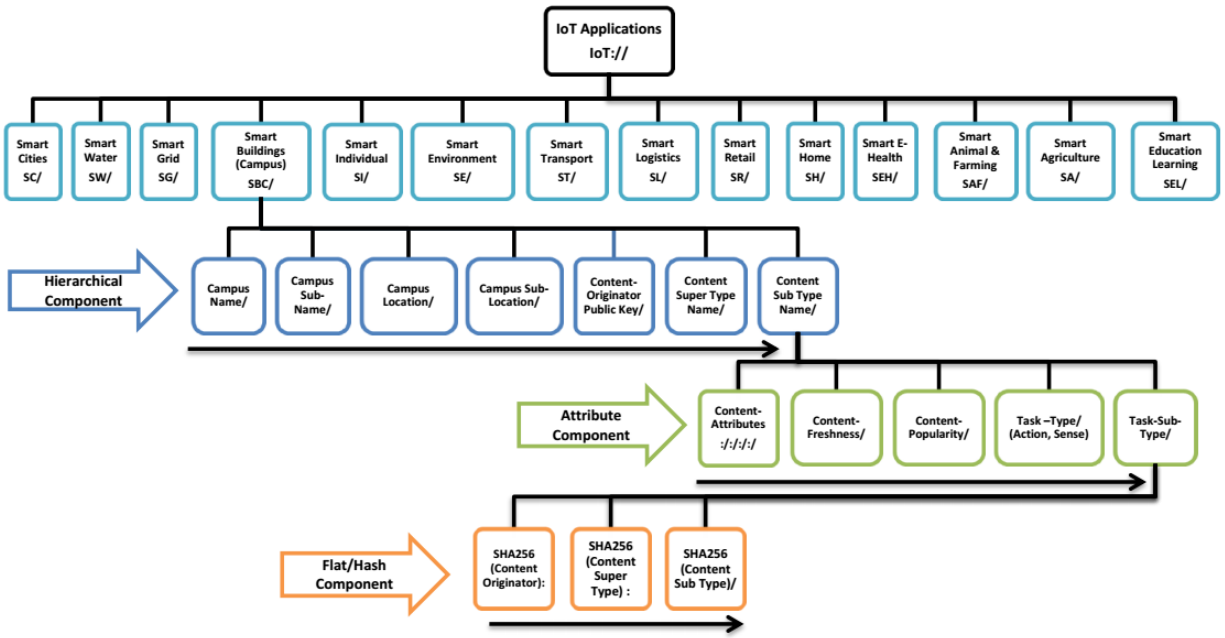

Figure 5.3: IoTs Applications Naming and Resolution. Step 1: Hierarchical Part and Resolution. Step 2: Attribute Part and Resolution. Step 3: Flat/Hash Part and Resolution.

with the physical location of campus instead it has just to represent name of location and sub-location of campus. HC's content information (both super type and sub type) is added to identify content main category and sub category. So that, HC can provide easy name management and aggregation, simple and easy search, and optimized routing tables by including above mentioned information about campus and content. HC part's information is described as follows:

- **Campus Name/Campus Sub-Name/:** This refers to the name of campus, for example, in our scenario, UET Taxila/CPED values refer to the Computer Engineering Department (CPED) of University of Engineering and Technology (UET), Taxila.

- **Campus Location/Campus Sub-Location/:** represent campus's country and city. In other words, where this campus is situated (or about which campus content is required), e.g., Pakistan/Taxila represents CPED is located in Taxila city of Pakistan.

- **Content Originator ID/Public Key:** This may involve student's registration number/faculty's employee name (or/and number), or sensor/actuator ID in case of devices e.g., 14F-UET-PhD-CP-43 represents registration number of Computer (CP) student and can refer to her/his cell phone device.

- **Content Super-Type Name/Content Sub-Type Name:** represents the content's super type from *text, image, video etc.* and sub type from *.word, .txt, .ppt etc.* for text, *.jpg, .png etc.* for image and *.avi, .mp4 etc.* for videos. For instance, *Timetable-14CP/.xls* refers to the text file of Timetable-14CP in .xls (and not in any other format e.g., .word, .txt, .ppt). This ensures further the exact type of the required content. In addition, by this, required content can be searched easily.

Moreover, HC's sub parts can be clearly seen in the second line (in ink-blue color) of Fig. 5.3.

5.3.3.2 Attributes Component (AC)

This component describes the details about a specific content. Attributes include content properties and the task type. AC's both task type and task sub-type specify type of communication that either sensing operation information is required or it will trigger/control some action. Here one thing is important to note that, inherent NDN supports only pull type communications. But with our naming scheme, push type communication will also be carried. Details of AC's sub-parts in following text:

- **Content-Attributes:/:/:/:** holds the values of multiple attributes for the desired content to signify it further, for example, *14/01-Jan/13:30/1/* refers to the extracted content's year-session (2014), date (1st January), time (13:30) and version (01) according to attributes *session/date/time/ver/*.

- **Content-Freshness/:** tells about content's life. One can specify about content's freshness in this, may be, by adding the time-stamp of content's generation time. IoTSC subscriber can get most recent and updated value by specifying this value as zero. And most old value can be get by specifying as one. This feature is added especially for sensors that provides some measurements (e.g., temperature, humidity, etc.) to enable pure IoT functionality like

getting the most recent value of temperature in campus. This can further help in content caching decision, which is currently beyond the scope of this work.

- **Content-Popularity/:** specify the popularity of the content. For instance, if a exam result is declared, then every student must be interested to get it. Contents can be easily be accessed (and provided) from near-by devices (even from friend's cell phone). Moreover, course video lectures marked as popular can be accessed more quickly. Its a simple counter that increments itself on every request. Moreover, other complex methods (e.g., Zipf's Distribution) can be explored to calculate popularity that helps in caching decision.

- **Task-Type/Task-Sub-Type:** holds the values about what is going to happen. Both *Task-Type/Task-Sub-Type* specify which operation will happen. Whether it is some action with the values *action/Turn-Light:(ON or OFF)* or sensing operation with the values *sense/(Temperature or Humidity)*.

Moreover, AC's sub parts can be clearly seen in the third line (in light green color) of Fig. 5.3.

5.3.3.3 Flat Component (FC)

This part is included to provide secure, signed and self-certified contents names. FC can hold the hashed value of either Content Originator ID or Content Super-Type Name or Content Sub-Type Name

- **SHA256(Content Originator):/SHA256 (Content Super-Type): / SHA256(Content Sub-Type):/ :** embeds hashed encrypted values of content originator and of content's name. For example,968cbab1de...:/e95e2bf0247.../0ac8b624229a...:/ represents SHA256s for 14F-UET-PhD-CP-43/Timetable-14CP/.xls.

Moreover, FC's sub parts can be clearly seen in the last line (in Orange color) of Fig. 5.3. However, one can argue why to use SHA256, as it creates 32-bytes data even for a small content like *.xls*. Therefore, one may use Base64 format to create FC. Study and selection for suitable creation methods for FC are part of future work.

5.4 Proposed NDN for IoTs

This section provides the description of NDN-HNS from our previous work [182]. Then details of NDN-HNFS are described through IoT traffic type and support for MTC.

5.4.1 Description of NDN-HNS for IoTs and Traffic Types

We take IoT-based Smart Campus (IoTSC) as a scenario to describe proposed NDN-HNS, which is a true representative scenario of IoT due to its need for scalability, heterogeneity, security and MTC. It can be seen name components of proposed NDN-HNS in Fig. 5.3. NDN-HNS incorporates hierarchical naming as hierarchical component (HC), attribute naming as attribute component (AC) and flat naming as flat component (FC). In short, NDN-HNS can be described as *HC:AC:FC:/*. HC is arranged to include general properties of application and content which can be used as prefix. HC include information about device, content name and type besides application information (upper portion of Fig. 5.3 against HC). These all sub-parts of HC is separated through '/' and ended by ':'. Then through AC we incorporate properties of content and MTC information. Content properties include general properties like date, time, version separated through ':/' and specific properties like freshness, popularity and MTC information include task type and task sub-type values separated through '/'. (middle portion of Fig. 5.3 against AC). Then lastly FC, which is added with the purpose to provide security include hashed-values of device-information, content name and its type are appended and separated through ':/'. (lower portion of Fig. 5.3 against FC). NDN-HNS include name components which later helps in finding the content on the basis of attributes. Further, more details of these three components can be found in [182].

Regarding working of NDN-HNS, when any NDN-IoT consumer sends request as *interest message (im)* for any content object towards nearby node, the node can reply from its own CS or forward this *im* towards other related nodes. Whenever any NDN-IoT node (which include producer, consumer and content routers (CRs)) receive any message it process this message on the basis of Algorithm 1 and then it calls Algorithm 2 to determine message type. Procedure to determine message type takes two values of Task-Type and Task Sub-Type from Algorithm 1 in the variables

Table 5.2: Representative Cases for All the IoT Traffic Possibilities

NDN-HNS
CampusName/CampusSubName/CampusLocation/CampusSubLocation/ ContentOriginator-ID/ContentSuperTypeName/ContentSubTypeName/: ContentAttributes:/:/:/:/ContentFreshness/ContentPopularity/TaskType/TaskSubType/: SHA256(Content Originator):/SHA256(Content Super-Type):/ SHA256(Content Sub-Type):/
Case: Pull (PLTC)
Case 1 Interest Message: Student is interested to retrieve Network, Assignment 2, UET/CPED/Pakistan/Taxila/14F-UET-PhD-CP-43/14CPNetworkAssignemnt,2/.xls/: 17:/0125 :/1330:/01:/0/0/Action/Retrieve/: 968cbab1de...:/9f087fb8a1cc6...:/0ac8b624229a...:/ Case 2 Interest Message: Student is interested in temperature Value (may be of Network Lab) UET/CPED/Pakistan/Taxila/14F-UET-PhD-CP-43/Temperature/NetworkLab/: 17:/1202 :/1100:/01:/0/0/Sense/Retrieve/: 968cbab1de...:/b958ce8b87...:/94341d04c8...:/
Case: Push (PHTC)
Case 3 Interest Message: Admin Employee is interested to turn-off AC1 of Network Lab UET/CPED/Pakistan/Taxila/ EMP-ADM-778/NetworkLab/NL-AC1/: 17:/0327 :/1520:/01:/0/0/Action/TurnOff/: 80ce94778...:/94341d04c8...:/f6f7ae2e808...:/ Case 4 Interest Message: Admin Employee is interested to set temp. of AC1 of Network Lab UET/CPED/Pakistan/Taxila/ EMP-ADM-778/ Temperature /NL-AC1 /: 17:/0815 :/1100:/01:/0/0/setValue/26/: 968cbab1de...:/b958ce8b87...:/f6f7ae2e808...:/ Case 5 Interest Message: Teaching Employee is interested to Upload Network Assignment-2 UET/CPED/Pakistan/Taxila/ EMP-TECH-980/ 14CPNetworkAssignemnt 2/.xls /: 17:/1202 :/1100:/01:/0/0/Action/Transmit-upload/: 108b4fb6...:/9f087fb8a1cc6...:/0ac8b624229a...:/ Case 6 Interest Message: When a sensor NL-AC1 send a periodic notification (say updated Temperature Value) UET/CPED/Pakistan/Taxila/NL-AC1/Temperature /26 C/: 17:/0815 :/1100:/01:/0/0/Action/Update/: 968cbab1de...:/b958ce8b87...:/f6f7ae2e808...:/

Case 7 Interest Message: When a sensor have to send an event-triggered notification (say Temperature value exceeds Threshold) UET/CPED/Pakistan/Taxila/ NL-TEMP1/Fire!/Exit /: 17:/0624 :/1100:/01:/0/0/Action/Trigger/: 108b4fb6...:/9f087fb8a1cc6...:/0ac8b624229a...:/
Case: Data Message
Case 8 Data Message: Student received data for network assignment 2 UET/CPED/Pakistan/Taxila/14F-UET-PhD-CP-43/14CPNetworkAssignemnt-2/.xls/: 17:/0125 :/1330:/01:/0/0/Action/Receive/: 968cbab1de...:/9f087fb8a1cc6...:/0ac8b624229a...:/[Content]

of action and subAction to decide message type. With the identified message type, every NDN-IoT node process it according to some constraints. These constraints include OnboardICNg by nodes when request is about to perform some action or when subscribed update data is received from CR which needs authentication and integrity checking to see whether its coming from the said device. We call this small algorithm as OnboardICNg. This method can be called by producer, consumer and CR during the processing of messages.

5.4.2 Description of NDN-HNFS for IoTSC

Next, we proposed forwarding algorithms to support IoT traffic types which we mention in Table. 5.2 and call NDN based hybrid naming and forwarding scheme as NDN-HNFS. As can be seen in Table. 5.2, IoTSC includes both PusH Type Communication (PHTC) and PulL Type communication (PLTC) to ensure handling of all requests which are categorized through first seven cases (i.e., see case 1 to case 7 in Table. 5.2).

5.4.2.1 Case: PLTC

In IoTSC, PLTC is dedicated to handle the situations when any subscriber from students, teaching staff and admin staff is interested to fetch some content and can be seen as case 1 and 2 in Table. 5.2. For instance, as case 1 we deal students who are usually interested to retrieve course assignments, test notification, etc. And for case 2, although the real users are from administration staff, who need to sense the temperature values to perform some other task lets say to set the value of any air

NDN-HNS Message Processing for IoTSC LS (Lab Sensors), CAS (CAmpus Server), mobile devices (MD) and LAS (LAb Severs i.e., Wi-Fi)

Possible input message: Incoming message.
Possible output message:*im*, *adv*, *dm*, *um*, μm, *bm*.

```
1  while last ':/' do
2      if '/' encounters then
3          save value before / in array[i] and update array[i+1]
4          continue scanning → goto while
5      else if '/:' encounters then
6          if first '/:' encounters then
7              save values of array[i-1] & array[i-2] & arr[i-3] into contentName and
                contentSubName and contentOriginatorID respectively
8              continue scannaing → goto while
9          else if second '/:' encounters then
10             save value of array[i-1] & array[i-2] into action and subAction respectively
11             Goto Message Type Determination and get messageType
12             and then continue scanning → goto while
13     else if ':/' encounters then
14         if first ':/' encounters then
15             save value of array[i] into year
16             continue scanning → goto while
17         else if second ':/' encounters then
18             save value of array[i] into month and date
19             continue scanning → goto while
20         else if third ':/' encounters then
21             save value of array[i] into time
22             continue scanning → goto while
23         else if fourth ':/' encounters then
24             save value of array[i] into version
25             continue scanning → goto while
26         else if fifth ':/' encounters then
27             save value of array[i] into shaOriginatorID
28             continue scanning → goto while
29         else if sixth ':/' encounters then
30             save value of array[i] into variable shaContentSuperName
31             continue scanning → goto while
32         else if seventh ':/' encounters then
33             save value of array[i] into variable shaContentSubName
34             continue scanning → goto while
35         end
36     end
37 end
```

Algorithm 1: Algorithm for NDN-HNS

Procedure 1: NDN-HNS Message Type Determination Processing for for IoTSC LS (Lab Sensors), CAS (CAmpus Server), mobile devices (MD) and LAS (LAb Severs i.e., Wi-Fi)

Possible Inputs: *task and task sub-type* in the form of *action and subAction*
Possible output message: *im*, *adv*, *dm*, *um*, *cm*, *bm*.

```
Switch action & subAction
Case: action & retrieve || sense & retrieve
    return im
Case: action & turn
    return cm
Case: set-value & float-value
    return cm
Case: action & upload
    return adv
Case: action & update
    return um
Case: action & trigger
    return bm
Case: action & receive
    return dm
    return messageType
```

conditioner of any laboratory (or classroom). For example, let's say administration employee with it's employee number of EMP-ADM-778 as device (Mobile phone) ID is interested to know the temperature value of Air conditioner 1 of NetworkLab (NL-AC1). However, students (or teachers) can also become interested to know the current temperature value, which in return, can be fetched from any nearby temperature sensor.

In case of PLTC, IoTSC consumers (or subscribers) can issue Interest message and content can be obtained through legacy NDN (See Algorithm 3). However, proposed naming schemes helps in fetching the content object with some properties like freshness, version and popularity. Producer (See Algorithm 3) or CR search CS with above mentioned properties to fetch content object. In case of failure, the message is forwarded to meet the content properties.

5.4.2.2 Case: PHTC

As IoTs real power lies in handling MTC where machines (devices) can talk to each other. Therefore, we addressed MTC by enabling PHTC style along with PLTC. PHTC is discussed for the three situations for different IoTSC users (teaching employee, administration employee, student) and with different queries/tasks/interests. For example, case 3: when administration employee with it's employee number of EMP-ADM-778 as ContentOriginator (Mobile phone) ID is interested to turn off Air conditioner 1 of NetworkLab (NL-AC1), case 4: when administration employee with it's employee number of EMP-ADM-778 as ContentOriginator (Mobile phone) ID is interested to set the temperature value of Air conditioner 1 of NetworkLab (NL-AC1) and case 5: when teaching employee with it's employee number of EMP-TECH-980 as ContentOriginator ID is interested to upload (transmit) a file with name NetworkAssignment 2.xls. Case:6 when a sensor need to send periodic update towards subscribed admin employee through general interest for further monitoring, case:7 when a sensor with its sensor ID need to inform about any disaster event (for example, about fire or earthquake notification).

In case of PHTC, IoTSC consumers (See Algorithm 3) are either interested to perform some action or IoTSC producers (lab sensors) need to send content object as a result of some event. Both consumer and producer have ability to send and receive at any moment but some constraints are applied before consumer can perform action on producer (See Algorithm 3) (sensor) or consumer can data from producer without any request. This involves OnboardICNg which authenticates NDN-IoT node to perform specific action.

5.4.2.3 Case: Data Message

When any intermediate node or data consumer (e.g., student, admin employee or teacher) receive content object from any sensor or intermediate node. Our algorithm 3 which is designed for content router and consumer on the identification of message type as *data message (dm)* checks for OnboardICNg. If the *dm* matches and passes OnboardICNg then the message is *update message(um)* which is received as a result of subscription to specific producer. In case of OnboardICNg fails, it means the *dm* is received against any *im*.

NDN-HNS Processing for IoTSC Producers such as LS (Lab Sensors)

Possible events: Initialize adv, arrival of interest *im* from consumer *C=CAS,LAS,MD* for chunk *c*, periodic update ρu at time t to C, occurrence of critical event ξ near *LS*.
Possible output message: *adv*, *dm*, *um*, *cm*, *bm*

```
case: event
when initialize:
    broadcast adv each non-critical content object CO
when ξ; C:
    construct beacon bm for ξ
    broadcast beacon bm toward CAS,LAS,MDi...j
    create and send chunks c_1, ..., c_N toward CAS,LAS,MDi...j
when message is received from LAS, LSi or C
    check for message type im, dm, adv, cm, um or bm
    call NDN-HNS Message Processing and get messageType
    switch messageType
when im; C; c:
    send c to C in dm
when cm; C:
    perform OnboardICNg() through content originator ID
    if (matches) then
        perform action and send status of action asked in dm
    else
        drop um
    end
when ρu:t;C:
    construct um w.r.t. ρu according to subscribed ID C
    create and send um towards the subscribed ID C
```

Algorithm 2: Forwarding Algorithm for NDN-IoT Producer

Procedure 2: NDN-HNS OnboardICNg for IoTSC Producers such as LS (Lab Sensors) and CR

Possible inputs:contentOriginatorID, shaOriginatorID, contentName, shaContentName, contentSubName, shaContentSubName .
Possible output message: message authorized or not.

```
calculate sha256 of contentOriginatorID and compare with shaOriginatorID
if matches then
    calculate sha256 of contentName and compare with shaContentName
    calculate sha256 of contentSubName and compare with shaContentSubName
end
Check for contentOriginatorID in authorized list
if matches one then
    authorize message
end
```

NDN-HNFS Processing for IoTSC Content Routers (CR) e.g.,CAS (CAmpus Server) and LAS (LAb Severs i.e., Wi-Fi)

Possible events: Arrival of *adv*, arrival of interest *dm* from producer *P=LS* for chunk *c*, periodic update *um* for event ρu at time *t* from producer *P=LS*, arrival of *bm*.

Possible output message: *im*, *dm*, *cm*, ρu, *um*, *bm*

```
case: event
when message is received from LAS, LSi or C
    check for message type im, dm, adv, cm, um or bm
    call NDN-HNS Message Processing and get messageType
Switch messageType
Case: messageType advertisement adv is received from LSi
if need content then
    send interest message im or ρu with t towards LSi
Case: messageType dm is received from LAS, CAS, LSi
    if entry exists for dm in PIT then
        perform OnboardICNg() through content originator ID if matches then
            apply cache mechanism only if necessary
        else
            Apply cache mechanism
            Forward dm towards next appropriate node
        end
    else
        Forward dm towards next appropriate node and drop dm
    end
Case: messageType is im
    check CS if freshness == 0 & popularity ≥ 1
    if found chunk then
        Reply with chunk
    else
        create entry in PIT
        forward im to next neighbor if necessary
    end
Case: messageType is um
perform OnboardICNg() through content originator ID
if matches then
    create temporary Interest
    create entry in PIT
    apply cache mechanism if necessary
else
    Forward um towards next appropriate node
end
Case: Message type is bm
    create Mock Interest
    create entry in PIT for c_1, ..., c_N
    apply cache mechanism and forward if necessary
when cm; C:
drop and forward cm
```


Algorithm 3: Forwarding Algorithm for NDN-IoT Consumer and Content Routers

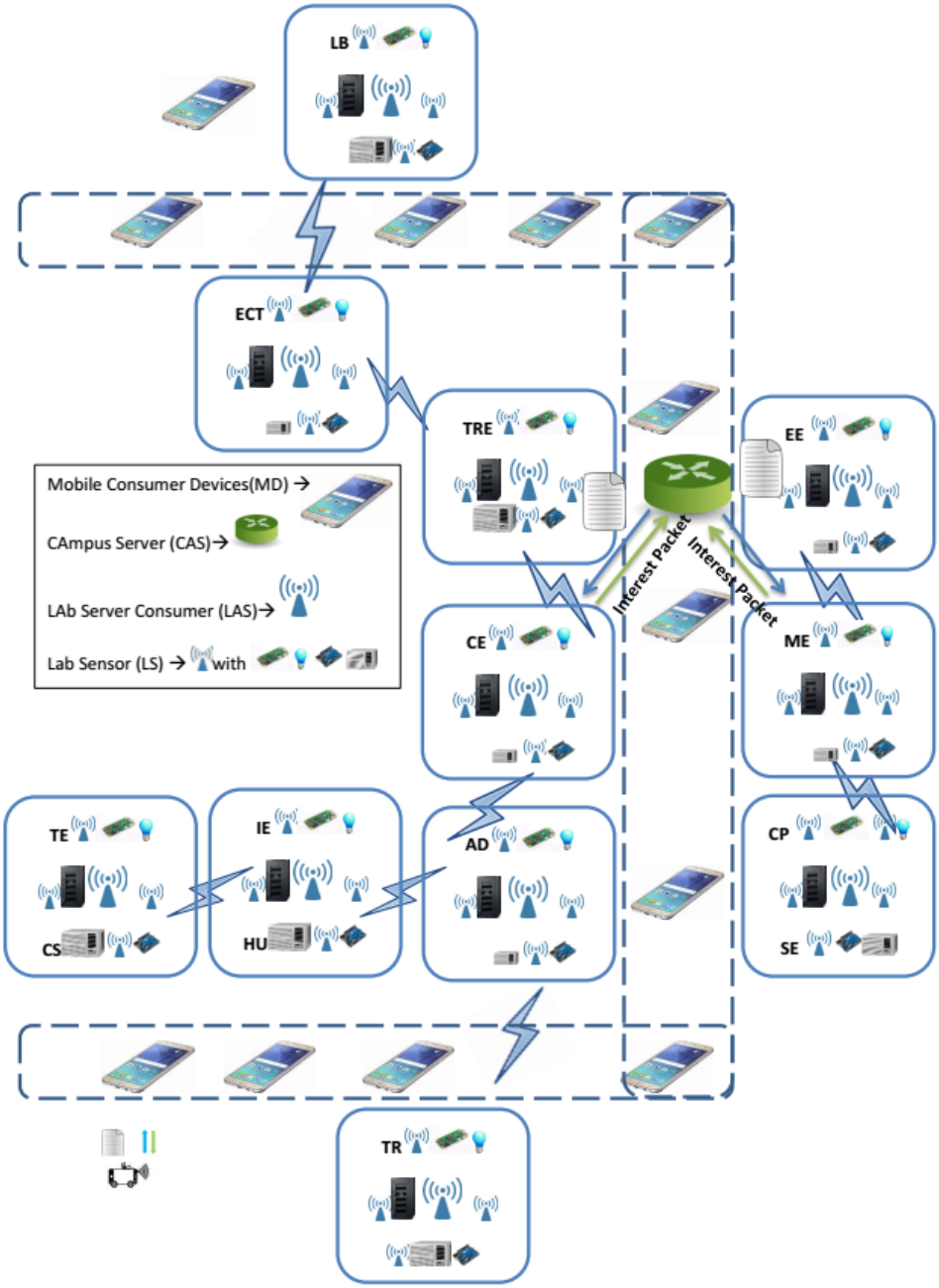

Figure 5.4: IoT True Representative Scenario: IoT-based Smart Campus.

5.5 Simulation and Evaluation of NDN-HNFS

5.5.1 Implementation Details

To implement proposed naming and forwarding scheme, we used ndnSIM which is NS-3 based NDN simulator. Implementation of NDN-HNS involves modified TLV files, both interest and data packet files and, modified consumer and producer files.

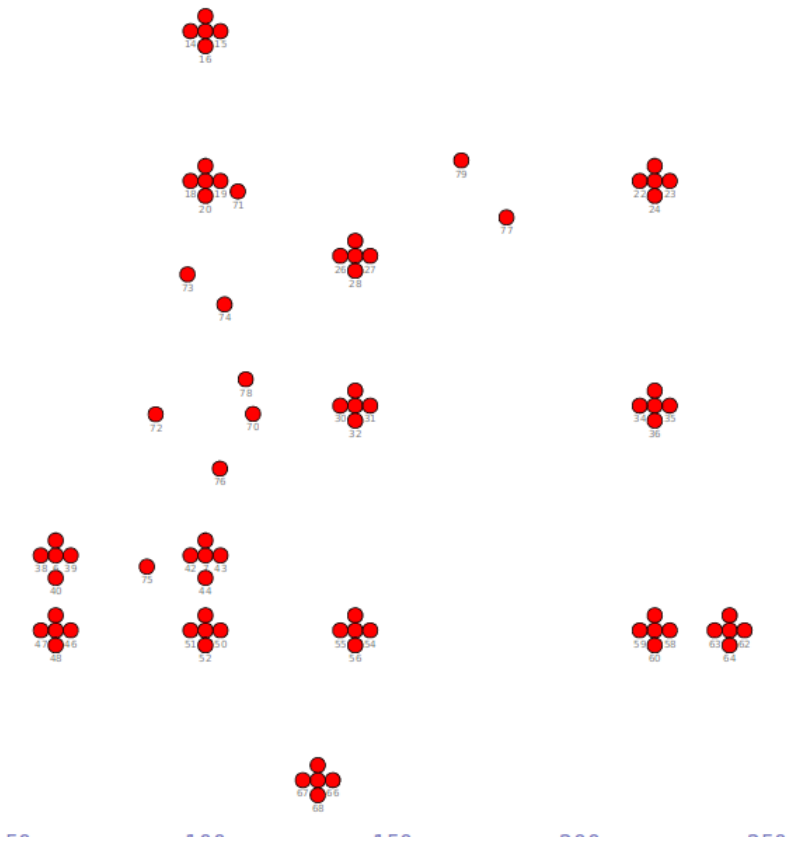

Figure 5.5: Simulation Graph for Educational and Non-Educational Departments of UET Taxila Generated from ndnSIM.

We added multiple new fields in interest and data packets according to Fig. 5.3. To evaluate proposed schemes, we design two scenarios. In first scenario, we implement native NDN by using ndnSIM functions. Then, in second scenario, we extend native NDN by adding proposed schemes. We calculate interest satisfaction rate (ISR), delay and number of exchanged messages for both scenarios.

5.5.1.1 Simulation Environment

We installed ndnSIM 2.0 on a system with Linux Ubuntu 14.04 LTS (64-bit) OS through dual booting. We assigned processor of Intel Core i5-2410M CPU 2.3GHz and 4GB memory to Ubuntu OS.

5.5.1.2 Scenario Description

Scenario of IoT-based Smart Campus (IoT-SC) is simulated in ndnSIM which is presented Fig. 5.4. To simulate it, we employed ndn over layer-2 IEEE 802.11 (WIFI-

Table 5.3: Simulation Parameters

Parameter	**Value**
Area	250×250
MAC Layer	Wi-Fi IEEE STANDARD 802.11a
Communication Stack	NDN
Number of Nodes	70 Static, 50 Mobile (Variable)
Number of producer nodes	56
Number of consumer nodes	50
Number of CR nodes	14
Interest frequency	variable
Number of Transmissions	500
Node Type	Tmote Sky
Simulation Time	500 Sec
Mobility Model	ConstantVelocityMobilityModel RandomWalk2dMobilityModel

PHY-STANDARD-80211a) nodes. Number of nodes are varied from total 80 to 130. Out of these total nodes, 70 nodes are static and mobile nodes are varied from 10 to 60. Mobile nodes are included to present human mobile consumers as ndn consumer application is installed on these consumer nodes. Simulation topology is shown in Fig. 5.5. Moreover, out of static nodes, 64 nodes are designated as producers and NDN producer application is installed on these producer nodes while rest of these static nodes are placed as content routers. Both *ConstantVelocityMobilityModel* and *RandomWalk2dMobilityModel* are employed for static and mobile nodes respectively. When any mobile consumer node request any content, the desired content is harvested from producers with the help of nearby CRs. We simulate this scenario for both native NDN and NDN-HNS for 500 seconds. Simulation Parameters are presented in Table 5.3.

5.5.2 Results and Discussions

The performance of proposed scheme NDN-HNFS is measured against legacy NDN in terms of delay and overhead through number of transmissions (or number of exchanged messages). The performance is measured and presented for both cases of PLTC and PHTC. We measure performance for every action mentioned in Table 5.2.

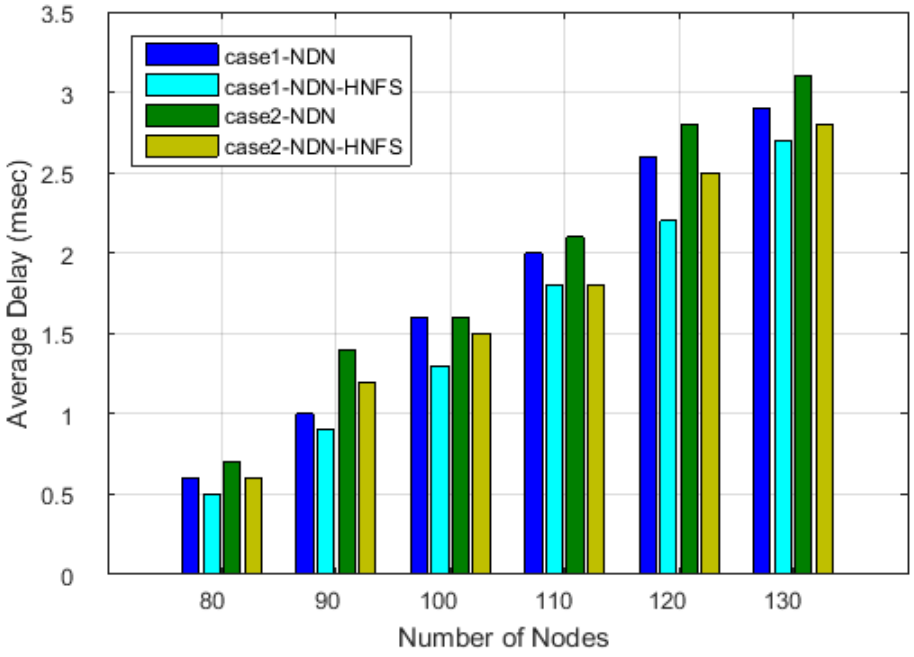

Figure 5.6: Delay for Case: NDN-HNFS-IoTSC-PLTC

5.5.2.1 Delay

Delay is the time difference between when an request is sent towards producer (or consumer) and *response* is received by consumer (or producer). Delay result for case of PLTC is visualized in Figure 5.6. It can be inferred from the delay result that for case of PLTC both NDN and NDN-HNFS exhibits almost same behavior but NDN-HNFS gives less delay as compared to legacy NDN. This is due to the reduction in time of interest generation, processing and responding.

Results for case PHTC are presented in Fig. 5.7 for case 3 to case 7. It can be seen that legacy NDN exhibits more delay than NDN-HNFS for case 3, 4, 6 and 7 but for case 5 which is uploading case incurs less delay for NDN in comparison.

5.5.2.2 Number of Exchanged Messages

It can be visualized from Figure 5.8 that average number of transmissions of case 3, 4, 6 and 7 against NDN-HNFS are less than legacy NDN. This is because legacy NDN does not support situations like case 3, 4, 6 and 7 and incurs more number of transmissions. Moreover, for case 5 NDN performs better than NDN-HNFS. This is due to the fact that NDN simply sends advertisement message to perform case 5.

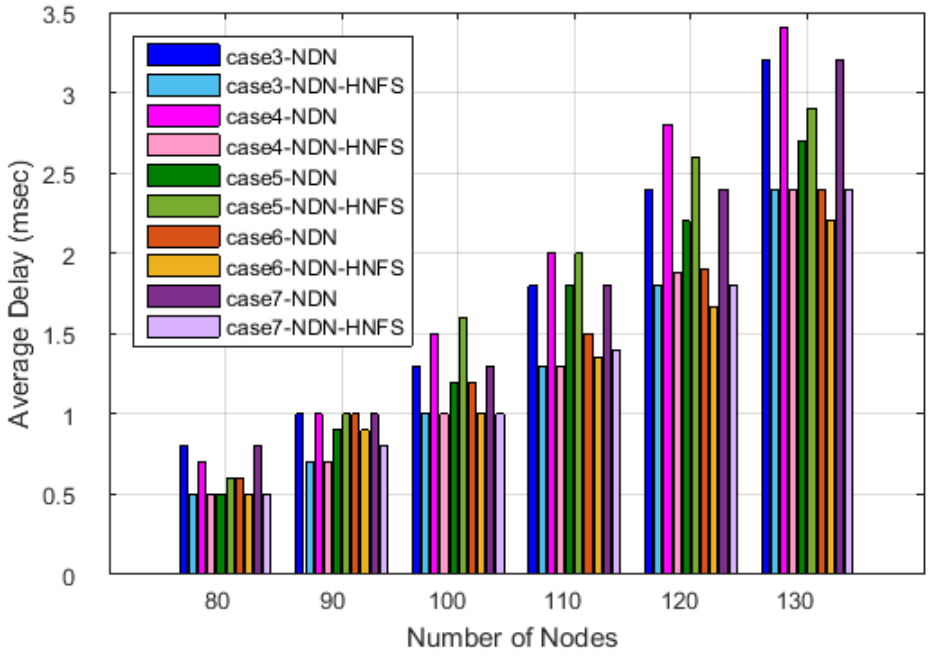

Figure 5.7: Delay for Case: NDN-HNFS-IoTSC-PHTC

5.6 Chapter Summary

We study ICN naming schemes for IoTs smart environment like smart home and smart campus. We proposed a sophisticated ICN-based (specifically NDN) naming scheme NDN-HNS for IoTSC. Moreover, we provide updated categorization of IoT applications. Proposed NDN-HNS operates in three important steps to assign a name to any content. As a first step, HC holds value about campus, content originator/requester and content itself. In second step AC contributes content's attributes, freshness, popularity and about task type. Security and authentication of the content is added by third component i.e., FC. On the basis of NDN-HNS, we listed IoT traffic types into seven cases for IoTSC and two communication models PLTC and PHTC to address pull and push type communication respectively. Seven cases for IoT traffic type, include almost every activity which can be carried in any IoT application, for example, update message, critical message, performing action on some device and setting the value of some parameter which (may be) is being sensed by device. Every IoTSC node executes NDN-HNS which in return, calls message type determination algorithm which is designed to identify message type among *bm, im, dm, um,* μ

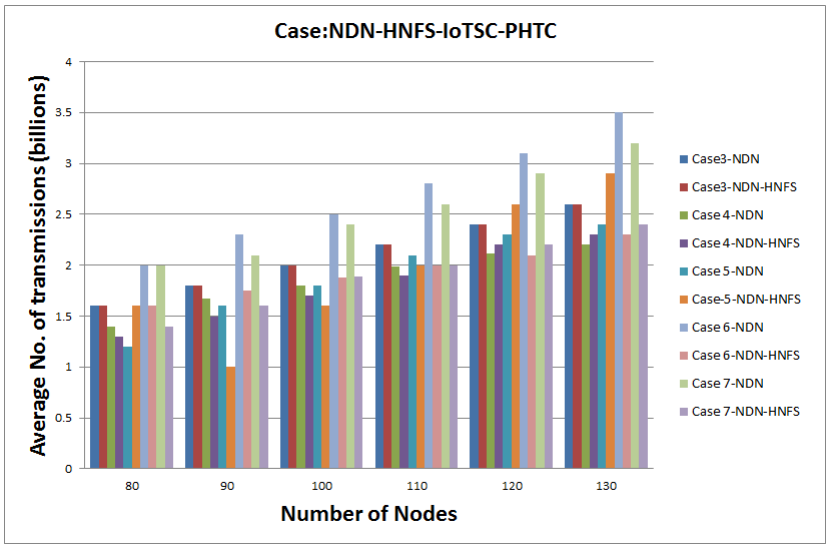

Figure 5.8: Overhead Transmissions for Case: NDN-HNFS-IoTSC-PHTC

is also associated with NDN-HNS. Then on the basis of NDN-HNS, we design forwarding algorithms (NDN-HNFS) for IoTSC nodes. Forwarding of messages involves OnboardICNg to ensure security in terms of authentication and authorization. We simulated UET Taxila smart campus in ndnSIM and evaluated NDN-HNFS against legacy NDN. We found NDN-HNFS more optimal one for IoTSC in terms of delay and number of transmissions.

We look forward to design a caching mechanism on the basis of proposed hybrid naming scheme. Moreover, we aim to emulate proposed schemes on hardware including real IoTSC devices like mobile phones, campus server, lab servers and lab sensors.

Chapter 6

Enabling Internet of Things with Efficient Named Data Networking based Forwarding

Internet of Things (IoT) has aimed to provide global access to information through connectivity among all devices. Also IoT is bringing all stand-alone smart applications under a single umbrella. Information-Centric Networking (ICN) or more specifically Named Data Networking (NDN) is recently being explored for IoTs and identified as a promising network architecture as contrary to traditional IP-based approaches. However, IoT needs both pull and push communication styles to enable machine-to-machine (M2M) functionality but legacy NDN only supports pull traffic. Moreover to provide global IoT connectivity, M2M security must also be maintained. Thus, we propose NDN-based holistic forwarding algorithms for both NDN-IoTs consumer and producer nodes that ensure security along with M2M communication support. We simulate the proposed schemes for IoT-based Smart Campus (IoTSC) in ndnSIM and found that our schemes result in overall better satisfaction rate as compared to legacy NDN schemes.

6.1 Introduction and Related Research Efforts

Named Data Networking (NDN) is a notable instance of Information-Centric Networking (ICN) and a future Internet architecture which is not only suitable for general data dissemination but also seems more appropriate for many IoT applications [183], one to mention here is smart building [109]. In NDN, each node maintains three data

structures namely Pending Interest Table (PIT), Forwarding Information Base (FIB) and Content Store (CS). Pending interests from different faces (interfaces) are stored in PIT while FIB keeps track of prefixes of forwarded interests towards producer nodes along each outgoing interface. The CS stores content data against each interest prefix and behaves as local cache of a NDN node. In the legacy NDN, communication only begins when an NDN consumer (e.g., a mobile node) issues an *interest message (im)* to receive the desired content named object. When any content router (CR) or a producer node receives *im*, it checks the content in its CS. If found, it replies with the specified content object in *data message (dm)*. It means, data (content object) is provided by publisher only if it is requested by the consumer and therefore it exhibits pull-based communication. Upon reception of any *dm*, consumer node checks for its prefix in the PIT and if found, only then *dm* is accepted and stored in the CS otherwise discarded while considering as unsolicited. NDN is proved to be a prominent architecture for IoT due to many benefits like elimination of middlewares to process data due to named contents and enhanced data availability through in-network caching [183]-[157]. Among many other challenges, for NDN-based IoT, only pull-based communication of NDN is not suffice because such an environment needs Machine-Type-Communication (MTC) to provide communication among IoT machines (i.e., devices). Therefore, to support MTC both push and pull modes are essential. In [121], four IoT traffic types are discussed and three push-support schemes are proposed for the identified traffic types. However, proposed schemes ignores the fact that by principle, any sensor can become producer and consumer at the same moment and included *command* in pull-based communication. Moreover, it requires an extra interest message from consumer to initialize push transmission which is infeasible due to resulting high traffic. Similarly in [184], authors propose forwarding algorithms for push traffic support in VANETs to send critical information and do not focus on IoT traffic types. All of these works ignored IoT security perspective. Thus, ICN (NDN)-based IoT (with MTC enabled) requires holistic data forwarding which is unsolved issue so far and is the main motivation behind this work.

In this chapter, to enable NDN-IoT, we propose forwarding algorithms for both consumer (content router) and producer which is mainly aimed to provide MTC with reasonable security and thus call it **NDN-IoT-SMTC**. This is achieved through

classification of representatives for all the MTC traffic possibilities into different types such as command, update and critical-data along with NDN traffic i.e., interest, advertisement and data. This classification is performed on the reception of any message using the name components *task* and *sub-task* and then the type of message is decided from the following: command (*cm*), beacon (*bm*), update (*um*), interest (*im*), data (*dm*) and advertisement (*adv*). The producer performs OnboardICNg [185] (which is a security mechanism) when it receives *cm* and as a response for *cm* while the consumer performs it when it receives *cm* or *um*. The proposed NDN-IoT-SMTC is designed to be applicable to any IoT application like smart home and smart hospital along with general NDN environments due to the features like, scalability, MTC, security, and content access through updated features such as freshness, popularity, and version of information. It is the first attempt to propose such NDN-IoT-SMTC to the best of our knowledge.

6.2 NDN-based IoT MTC-enabled Forwarding

To design data forwarding in NDN-IoT-SMTC, a NDN-based hybrid naming scheme (HNS) [182] is used to classify all traffic possibilities of IoT into different representative cases. NDN-based HNS names the contents through hierarchical component (HC), flat component (FC), and attribute component (AC) in a hybrid fashion. A complete content name is the combined form of HC:AC:FC (See Table 6.1). We take the scenario of IoT-based smart campus (IoTSC) to describe our work because IoTSC needs MTC (which needs both PulL Type Communication (PLTC) and PusH Type Communication (PHTC) styles) with security as IoT does. To support both the PLTC and the PHTC to form NDN-IoT-SMTC, we add two fields of **task-type** and **task sub-type** in NDN interest and data messages to identify message type. With the identified message type, PLTC and PHTC is proceeded further with the help of OnboardICNg security mechanism.

In OnboardICNg, firstly integrity of message is checked by calculating the hashes of message sender-ID, content name (CN), content-sub-name (CSN) using SHA256. After that the calculated hashes are compared with appended hashes of the FC. If all hashes are successfully matched, then sender-ID is checked to find a match in

Table 6.1: Representative Cases for All the IoT Traffic Possibilities

NDN-HNS
Country/State/City/UniName/UniSubName/ContentOriginator-ID/ DeviceRegistrationNumber/ContentSuperTypeName/ContentSubTypeName/: ContentAttributes:/:/:/:/ContentFreshness/ContentPopularity/TaskType/ TaskSubType/: SHA256(ContentOriginator-ID):/SHA256(ContentSuperTypeName):/ SHA256(ContentSubTypeName):/
Case: Pull (PLTC)
Case Query (im, ρu): When consumer (Student) is interested to retrieve content (Network Assignment2.xls) from network or interested to sense temperature value from any sensor (NetworkLab Sensor) or interested to subscribe for any content update 1: X/X/X/Uni/Dept./14F-Uni-PhD-CP-43/Samsung-J7/14CPNetworkAssignment-2/.xls/: 17:/0125:/1330:/01:/0/0/Action/Retrieve/: 968cbab1de..:/9f087fb8a1cc6..:/0ac8b624229a..:/ 2:X/X/X/Uni-Name/Dept./14F-Uni-PhD-CP-43/Samsung-J7/Temperature/NetworkLab/: 17:/1202:/1100:/01:/0/0/Sense/Retrieve/: 968cbab1de...:/b958ce8b87..:/94341d04c8..:/ 3:X/X/X/Uni-Name/Dept./14F-Uni-PhD-CP-43/Samsung-J7/Temperature/NetworkLab/: 17:/1202:/1100:/01:/0/0/Subscribe/Retrieve/: 968cbab1de...:/b958ce8b87..:/94341d04c8..:/ Case Command (cm): When an Admin (Employee) is interested to turn-XX or set value of any device (AC1 of Network Lab) 3.1:X/X/X/Uni/Dept./EMP-ADM-778/PC-311/NetworkLab/NL-AC1/: 17:/0327:/1520:/01:/0/0/Action/TurnOff/: 80ce94778..:/94341d04c8..:/f6f7ae2e808..:/ 4.1: X/X/X/Uni/Dept./EMP-ADM-778/Samsung-J7/Temperature/NL-AC1 /: 17:/0815:/1100:/01:/0/0/setValue/26/: 968cbab1de..:/b958ce8b87..:/f6f7ae2e808..:/
Case: Push (PHTC)
Case Command (cm): When any sensor (TEMP1 of Network Lab) is interested to turn-XX or set value of any device (AC1 of Network Lab) 3.2:X/X/X/Uni/Dept./NL-TEMP1/Arduino-Mega/NetworkLab/NL-AC1/: 17:/0327:/1520:/01:/0/0/Action/TurnOff/: 80ce94778..:/94341d04c8..:/f6f7ae2e808..:/ 4.2: X/X/X/Uni/Dept./NL-TEMP1/Arduino-Mega/Temperature/NL-AC1 /: 17:/0815:/1100:/01:/0/0/setValue/26/: 968cbab1de..:/b958ce8b87..:/f6f7ae2e808..:/ Case Update (um): When a sensor with its sensor ID (NL-AC1), send a periodic notification (say updated Temperature Value) 5:X/X/X/Uni/Dept./NL-AC1/Arduino-Mega/Temperature/26C/: 17:/0815:/1100:/01:/0/0/Action/Update/: 968cbab1de..:/b958ce8b87..:/f6f7ae2e808..:/

Case Event-triggered (bm): When a sensor with its sensor ID, send an event-triggered notification (say Temperature value exceeds Threshold) 6:X/X/X/Uni/Dept./NL-TEMP1/Arduino-Mega/Fire!/Exit/: 17:/0624:/1100:/01:/0/0/Action/Trigger/: 108b4fb6..:/9f087fb8a1cc6..:/0ac8b624229a..:/ Case Advertisement (adv): When any user (Teaching Employee) is interested to upload any content (NetworkAssignment-2) 7: X/X/X/Uni/Dept./EMP-TECH-980/Samsung-J7/14CPNetworkAssignment-2/.xls /: 17:/1202:/1100:/01:/0/0/Action/upload/: 108b4fb6..:/9f087fb8a1cc6..:/0ac8b624229a..:/
Case: Data Message
Case Data Message (dm): When any user (Student) receive data for queried content (Network Assignment) or from any sensor in response to cm X/X/X/Uni/Dept./14F-Uni-PhD-CP-43/Samsung-J7/14CPNetworkAssignment-2/.xls/: 17:/0125:/1330:/01:/0/0/Action/Receive/: 968cbab1..:/9f087fb8a1c..:/0ac8b62422..:/[Content]

authorized list which (we assume) is saved in the node memory during node initialization phase. If authorized list has an entry for sender-ID, only then the message is authorized to perform required action.

When any of the devices among Lab Sensor (LS), LAb Server (LAS), Mobile Device (MD) and CAmpus Server (CAS) receive any message (Table 6.1), they firstly call message processing function which calls message type determination function that identifies message type on the basis of task and sub-task. Values of task and sub-task are checked and compared to decide the message type. When values of task and sub-task are **action & retrieve (or sense & retrieve), subscribe & retrieve, action & upload, action & update, action & trigger, action & turn (set-value & float-value)** and **action & receive**, then message type is identified as *im, ρu, adv, um, bm, cm* and *dm* respectively. On the basis of these identified message types, every node process it with the specified rules using OnboardICNg.

a) Case: Producers (or Sensors)

For processing of the sensors or producers, refer to Algorithm 1. In PLTC (Table 6.1: case 1.1 to 1.3), when an IoTSC sensor (i.e., LS) receives *im* from other devices like MD, LAS or CAS for content chunk *c* of content object (CO), LS responds with chunk *c* through *dm*. To support PHTC four cases are designed when

Algorithm 1 NDN-IoT-SMTC Processing for an IoTSC LS (Lab Sensors)
Possible Events:[initialize adv, arrival of *im* or periodic update ρu at time t for chunk c, from consumer *C=CAS, LAS, MD*, arrival and transfer of *cm*, arrival of *dm* for *cm*, occurrence of critical event ξ near *LS*]
Possible Output Messages: [*adv*, *dm*, *cm*, *um*, *bm*]

1 case **event**
2 **When initialize:**
3 Broadcast *adv* for each non-critical content object *CO*
4 **When ξ; C:**
5 Construct beacon message *bm* for ξ
6 Broadcast beacon message *bm* towards *CAS,LAS,MDi...j*
7 Create and Send chunks *c1,...,cN* towards *CAS,LAS,MDi...j*
8 **When to perform some action on *LSj* (Only for local devices)**
9 Create entry in PIT
10 Send *cm* towards *LSj*
11 **When message is received from LAS, LSi or MD**
12 Call NDN-HNS message processing and Get message type (*im*, *dm*, *cm*, ρu)
13 **Case: When *im; C; c:***
14 Send *c* to *C* in *dm*
15 **Case: When *cm; C:***
16 Perform OnboardICNg through ID, CN and CSN
17 If matches
18 Perform action and send status of action asked in *dm*
19 else drop cm
20 **Case: When *ρu:t;C:***
21 Construct *um* w.r.t. ρu according to subscribed ID C
22 Create and send *um* towards the subscribed ID C
23 **Case: When *dm* for *cm* is received from LSj**
24 if entry exists for dm in PIT
25 Perform OnboardICNg through ID, CN and CSN
26 if matches then
27 Apply cache mechanism only if necessary
28 else drop dm

LS has to push some data without request (Table 6.1 case: 2 to 5). For example, whenever LS has normal non-critical CO (case:2), it sends *adv* for each CO. Further, for *um* (case:3) which is response to timely update ρu (case:1.3) for subscribed CO, LS constructs, creates and sends *um* for ρu using FIB in a similar way to *dm* (Table 6.1, last case of *dm*). Furthermore, when LS has critical or event-triggered data due to occurrence of critical event ξ (case:4), it (LS) construct, create and broadcast *bm*. And when any LSj has to perform action or change status of any other LSi just like consumer C (cases 5.1 & 5.2:*cm*), it (LSj) creates entry in the PIT and sends *cm* for this purpose. When message-type is identified as *cm* by any LSi, it is checked that sender (producer LSj or consumer C) is valid or not to perform *cm* through OnboardICNg. Upon successful authorization from OnboardICNg, LSi can perform action and generate required CO. Response or status of LSi is sent through *dm* towards consumer C or LSj using the FIB. Message *cm* is dropped if the authorization gets declined. Furthermore, upon reception of *dm* for *cm*, LSj checks in the PIT and performs OnboardICNg to accept or drop the *dm*. If the *dm* is accepted through OnboardICNg, it is cached if necessary.

b) Case: Content Router (CR) or Consumer

Forwarding details of C or CR and downstream of CR can be seen in Algorithm 2 and Fig. 6.1 respectively. In PLTC (Table 6.1:case 1.1 to 1.3), when CR gets message type identified as *im* through message determination then the CS is checked only when requested CO is popular but not fresh (older). Otherwise, entry is created in the PIT and forwarded to next node. On the successful search for *c*, the CO is sent towards consumer. Further for PHTC, when message is identified as *adv* (case:2), if C is interested in CO, it can create the PIT entry and send *im* for advertised *adv* or ρu to subscribe update in the CO.

Furthermore, when C receives *um* (case:3), CO is accepted after successful OnboardICNg otherwise it is dropped and forwarded to next node. To accept CO against *um*, mocked interests are generated through mock interest generator and are placed in the PIT. Mocked Interests are generated in a situation when unsolicited CO is needed to accept and store in CS. Then, either CO is cached in the CS or dropped on the basis of its existence or absence in the PIT respectively.

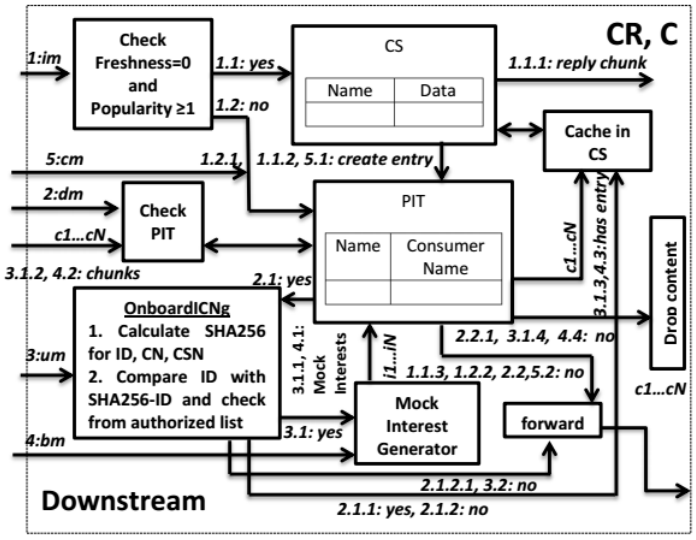

Figure 6.1: NDN-IoT-SMTC Data flow for Content Router (CR)

Moreover, when received message is identified as *bm* (case:4), then (just like *um*) mocked interests are generated for *bm* through mocked interest generator and are placed in the PIT. Upon reception of chunk *c*, both CR and C caches after checking its entry in PIT but only CR broadcasts *bm*. As case:5.1 to 5.2 of *cm* when any consumer, which is also administrator (from MD or desktop system) is interested to perform some action, such consumer creates the PIT entry for it and this entry is forwarded as *cm* towards LSi through next node. In addition, when CR (or LAS) identifies upcoming message as *cm*), CR simply forward the *cm* towards LSi. Finally, when the CR receives *dm* (case:6), PIT is checked for such entry. If PIT entry is found then OnboardICNg is performed, otherwise *dm* is forwarded to the next node and dropped. In case after successful OnboardICNg, CR can definitely cache CO if it is the response to *im* and can drop if it the response to *um*. If *dm* is failed in OnboardICNg, it means that it is not tentative consumer of *um*. Thus, CR has to cache and forward CO towards appropriate node with the help of FIB. In short, if *dm* is response to *um* and OnboardICNg is successful, only then CO is cached. Otherwise, it is the response to *im* and can be forwarded after caching.

Algorithm 2 NDN-IoT-SMTC Processing for IoTSC CAS, MD and LAS
Possible Events:[arrival of *adv*, arrival of *dm* from producer *P=LS* for chunk *c*, periodic update *um* for event ρu at time *t* from producer *P=LS*, arrival & transfer of *cm*, arrival of *bm*]
Possible Output Messages: [*im*, *dm*, *um*, *cm*, ρu, *bm*]

1 **case event**
2 **When to perform some action on *LSi***
3 Create entry in PIT
4 Send *cm* towards *LSi*
5 **When message is received from LAS, LSi or MD**
6 Call NDN-HNS message processing and Get message type (*im*, *dm*, *um*, *cm*, *adv*, *bm*)
7 **Case: message type is im**
8 Check CS if freshness==0 & popularity ≥ 1
9 if found chunk then
10 Reply with chunk in dm
11 else create entry in PIT and then send im to next neighbor
12 **Case: message type is dm for chunk *c* is received from LAS, LSi, CAS**
13 if entry exists for dm in PIT
14 Perform OnboardICNg through ID, CN and CSN
15 if matches then
16 Apply cache mechanism only if necessary
17 else apply cache mechanism and forward dm towards next node
18 else forward dm towards next node and drop dm
19 **Case: message type is um**
20 Perform OnboardICNg through ID, CN and CSN
21 if matches then
22 Create mock interest
23 Create entry in PIT
24 Receive chunks
25 Apply cache mechanism if necessary
26 else drop um and forward um towards next node
27 **Case: message type is bm**
28 Create mock interest
29 Create entry in PIT for *c1,...,cN* and receive chunks
30 Apply cache mechanism and broadcast if necessary (if CR)
31 **Case: message type is cm**
32 Create entry in PIT and then send *cm* to next node towards LSi
33 **Case: message type is adv is received from *LSi***
34 if need content
35 Create entry in PIT
36 Send interest message *im* or ρu with *t* towards *LSi*

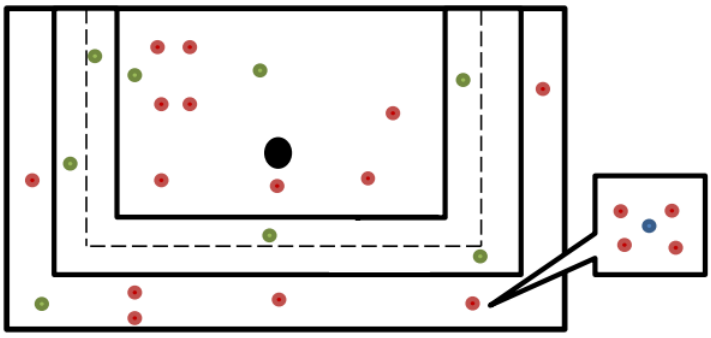

Figure 6.2: Simulation IoT Scenario: Red colored dots are for departments, black colored are for CAS (CR,C), green colored dots are for MD (C). Each department assumed to have five nodes (shown in small block with four red and one blue dot(s)), among four red dots are LS (P) and one blue dot is LAS (CR,C)

6.3 Simulation and Performance Evaluation

To evaluate NDN-IoT-SMTC, we simulate an IoTSC which consists of fourteen academic and non-academic departments of a university (See Fig. 6.2). Each department contains four producer (static nodes) which are placed through 'ConstantVelocity-MobilityModel' to sense the environment or actuators to actuate the commands on department devices such as Air-Conditioner (AC) or heaters and one static consumer node (LAS) which is also responsible to harvest or deliver information from/to other four nodes. A CAS static CR node (black dot in Fig. 6.2) is placed in the center of campus and variable number of MDs (green dots) are placed randomly through 'RandomWalk2dMobilityModel' to mimic the presence of mobile users which can be student, teacher, visitor or admin employee. Therefore, total 70 static nodes (14 LAS, 56 LS) and from 10 to 60 number of mobile nodes (MD), resulting in maximum nodes ranging from 80 to 130 nodes, are considered for IoTSC. All nodes are based on IEEE 802.11 (WIFI-PHY-STANDARD-80211a) Wi-Fi. We design total sixteen scenarios to evaluate the eight possibilities (1.1 to 1.3, 2 to 4 and 5.1 to 5.2) of first five cases (im, adv, um, bm, cm) which are discussed in Table 6.1. Every possibility of each case is evaluated on both proposed NDN-IoT-SMTC and the legacy NDN. Total area covered is 250×250m and simulations are run for 500 seconds. Every consumer node (LAS, MD) generates 5 to 10 interests per second. We save 100 content objects in CS according to freshness value.

The performance of proposed algorithms is measured against the legacy NDN

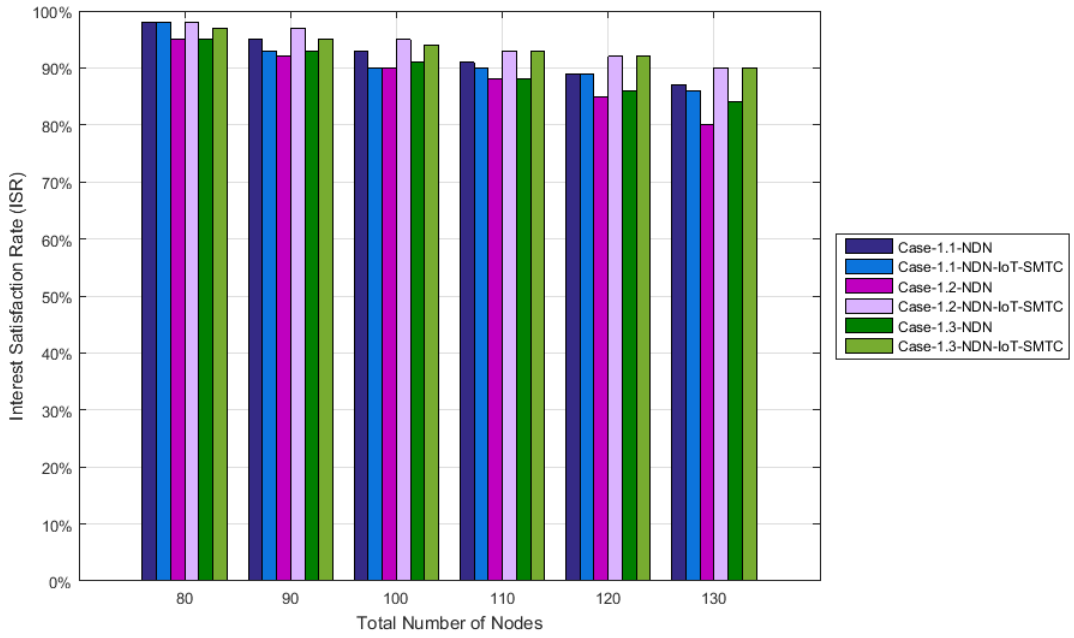

Figure 6.3: ISR for Case: NDN-IoT-SMTC for IoTSC-PLTC

in terms of Interest Satisfaction Rate (ISR) which is the percentage of the ratio of the number of satisfied interests to the number of interests issued by a consumer to get a content. The ISR results exhibit that both legacy NDN and NDN-IoT-SMTC produce almost similar results for Case 1.1, 1.2 and 1.3 in case of PLTC as can be seen in Fig. 6.3. The reason behind this that NDN natively supports such cases when a consumer is interested to fetch or subscribe data for which it can issue an interest message. PHTC ISR results can be seen in Fig. 6.4. For Case 2 legacy NDN and NDN-IoT-SMTC produce almost similar results because legacy NDN supports advertisement scenario. For Case 3 and 4 legacy NDN gives zero output as it does not support these type of cases. In contrast, NDN-IoT-SMTC gives 75% to 90% ISR. For Case 5.1 and 5.2, legacy NDN has approximately 40% less ISR than NDN-IoT-SMTC. In nutshell, NDN-IoT-SMTC outperforms legacy NDN and provides support for every action which any smart campus user requires. Moreover, it can be seen that ISR decreases when number of nodes increase and this is due to increasing number of transmissions and fixed simulation time.

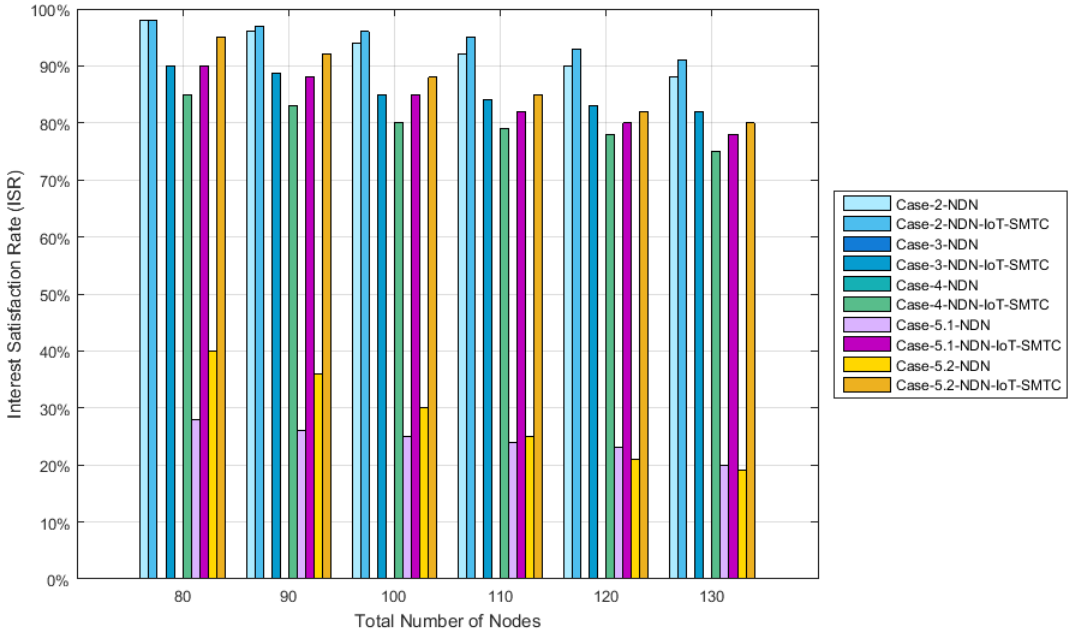

Figure 6.4: ISR for Case: NDN-IoT-SMTC for IoTSC-PHTC

6.4 Chapter Summary

We extensively study and classify IoT traffic types and find that it lacks holistic support which considers forwarding of every traffic type in NDN-IoT. Moreover, we identify that NDN-IoT lacks in providing efficient support for MTC in a secured way. Thus, we propose forwarding algorithms NDN-IoT-SMTC for both NDN producer and consumer nodes. To enable MTC in an efficient and secured way, producer node is enabled to support pushing data towards consumer through periodic update message, perform action message and beacon message. The relevant handling of these three different push messages is also updated and modified in consumer (or CR) nodes. The proposed schemes are implemented in ndnSIM and simulated for IoTSC which is a true representative scenario of IoT. Through results, it is concluded that NDN-IoT-SMTC shows improved ISR results than native NDN. As follow-up work, we aim to evaluate proposed forwarding algorithms on real-time IoT smart campus scenario using both NDN-Arduino and NDN-Android libraries on mobile phones and Arduino-based sensing devices.

Chapter 7

Conclusions & Future Work

We discussed and presented related literature of both new paradigms IoTs and ICN. We elaborated briefly IoTs four working phases namely: acquisition and sensing, data transmission, data processing and information management, action and utilization along with their corresponding technologies. Requirements and challenges to build a reliable and inter-operable communication network architecture for IoTs are presented. We have also discussed ICN suitable features, different ICN projects for the future Internet design and their resulting ICN based network architectures for IoTs. ICN projects are briefly discussed in terms of their corresponding feasibility for IoTs in terms of naming schemes, caching mechanisms, security and mobility support. Mapping of IoTs communication network architecture requirements against ICN striking and supporting features is presented. Furthermore, we discussed ICN based solutions/architectures for IoTs to present the applicability of ICN for IoTs. Then, we presented and classified ICN-IoT state-of-the-art naming literature into four categories of hierarchical, flat self-certifying, attribute-based and hybrid naming schemes.

Mainly, this thesis introduced the ICN (more specifically NDN (CCN)) for IoT and presented two novel naming mechanisms along with optimal forwarding scheme equipped with security, heterogeneity and scalbility. Apart from this, IoT applications categorization is also updated through fourteen categories. From updated categorization, IoT-based Smart Campus (IoTSC) scenario is selected to design naming and forwarding schemes due to its true representation for IoT.

As a basic scheme, CCN-based hybrid naming scheme is proposed which names the contents using hierarchical and flat components to support both *push* and *pull*

communication and introduced two transmission modes namely (1) unicast mode and (2) broadcast mode to address loop problem associated with CCN. The hierarchical component takes the domain name, location, task and device name in URL style. By using 'task', push support is added in the native CCN protocol. The flat component is used to provide integrity and it is computed through the FNV-1a hash of the device name and Data. The communication loop problem associated with CCN protocol is eliminated by implementing 'unicast' protocol on the source nodes. Mobile IoT nodes are used for delivery of Interest and Data packets to nodes that are not in the range of sink node. The proposed naming scheme is evaluated for IoT-SC having both static and mobile nodes and results revealed the significant gain in terms of satisfaction rate and number of transmissions of Interest packets, latency, number of hops and interest aggregation.

Further as an extended scheme, a sophisticated NDN-based hybrid naming scheme is proposed which names the IoT devices and content using hierarchical, flat and attribute components to support both *push* and *pull* IoT traffic models. Proposed NDN-HNS operates in three important steps to assign a name to any content. As a first step, HC holds value about campus, content originator/requester and content itself. In second step, AC contributes content's attributes like freshness, popularity and task type. Security and authentication of the content is added by third component i.e., FC.

Then on the basis of extended NDN-based naming scheme, IoT traffic types are defined using the listed activities in IoTSC. Holistic forwarding schemes are proposed for NDN-IoT consumer, producer and content routers which provide machine type communication (MTC) with push and pull communication models enabled. These forwarding schemes use another OnboardICNg security scheme which is designed to authenticate and authorize the devices to perform asked actions. These schemes enable NDN-IoT producer to send critical content or updates of subscribed content to NDN-IoT consumer through content router(s). Moreover, NDN-IoT consumer is enable to send message to perform any action or setting value of any parameter of NDN-IoT producer. These schemes are also implemented in ndnSIM and evaluated against legacy NDN in terms of interest satisfaction rate, latency and number of transmissions.

As a **future work**, we aim to evaluate proposed NDN-based hybrid naming scheme and forwarding algorithms on real-time IoTSC scenario using both NDN-Arduino and NDN-Android libraries on mobile phones and Arduino-based sensing devices (i.e., lab sensors) and Wifi based lab servers. Moreover, we look forward to design a caching mechanism on the basis of NDN-based hybrid naming scheme.

References

[1] Ierc-european research cluster on the internet of things. [Online]. Available: http://www.internet-of-things-research.eu/about-iot.htm

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in iot networking via tcp/ip architecture," NDN Project, Tech. Rep. NDN-0038, Tech. Rep., 2016.

[5] E. Borgia, "The internet of things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, no. 1, pp. 1–31, 2014.

[6] J. A. Stankovic, "Research directions for the internet of things," *Internet of Things Journal, IEEE*, vol. 1, no. 1, pp. 3–9, 2014.

[7] V. Varadharajan and S. Bansal, "Data security and privacy in the internet of things (iot) environment," in *Connectivity Frameworks for Smart Devices*. Springer, 2016, pp. 261–281.

[8] R. Silva, J. S. Silva, and F. Boavida, "Infrastructure-supported mobility in wireless sensor networks — a case study," in *Proc. IEEE Int. Conf. Industrial Technology (ICIT)*, Mar. 2015, pp. 1895–1900.

[9] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Impact of mobility on the iot MAC infrastructure: IEEE 802.15.4e tsch and lldn platform," in *Proc. IEEE 2nd World Forum Internet of Things (WF-IoT)*, Dec. 2015, pp. 478–483.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 4, pp. 2347–2376, 2015.

[11] (2014) Iot6.eu researching ipv6 potential for internet of things. [Online]. Available: http://www.iot6.eu/

[12] S. Ziegler, P. Kirstein, L. Ladid, A. Skarmeta, and A. Jara. (2015) The case for ipv6 as an enabler of the internet of things. [Online]. Available: http://iot.ieee.org/newsletter/july-2015/the-case-for-ipv6-as-an-enabler-of-the-internet-of-things.html

[13] ——. (2015) Understanding ipv6's potential for iot: The iot6 research project. [Online]. Available: http://iot.ieee.org/newsletter/september-2015/understanding-ipv6-s-potential-for-iot-the-iot6-research-project.html

[14] (2012) Ietf constrained restful environment (core) working group. [Online]. Available: https://datatracker.ietf.org/wg/core/charter/

[15] (2010) The constrained application protocol (coap). [Online]. Available: https://datatracker.ietf.org/doc/rfc7252/

[16] (2011) Ietf 6lowpan working group. [Online]. Available: https://tools.ietf.org/wg/6lowpan/charters

[17] (2008) Routing over low power and lossy networks (roll). [Online]. Available: https://datatracker.ietf.org/wg/roll/documents/

[18] (2009) Rpl: Ipv6 routing protocol for low-power and lossy networks. [Online]. Available: https://datatracker.ietf.org/doc/rfc6550/

[19] (2011) Ietf light-weight implementation guidance (lwig) working group. [Online]. Available: https://datatracker.ietf.org/wg/lwig/charter/

[20] (2015) Irtf thing-to-thing (t2trg) research group. [Online]. Available: https://datatracker.ietf.org/rg/t2trg/charter/

[21] (2017) Integrating objects to create new networked services. [Online]. Available: http://www.etsi.org/technologies-clusters/clusters/connecting-things

[22] (2017) Iot standards and protocols. [Online]. Available: https://www.postscapes.com/internet-of-things-protocols/

[23] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named data networking for iot: an architectural perspective," in *2014 European Conference on Networks and Communications (EuCNC)*. IEEE, 2014, pp. 1–5.

[24] (2012) Information-centric networking (icnrg). [Online]. Available: https://datatracker.ietf.org/rg/icnrg/about/

[25] Named data networking (ndn) project. [Online]. Available: http://named-data.net/

[26] Pursuing a pub/sub internet-fp7 project pursuit. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[27] Network of information (netinf). [Online]. Available: http://www.netinf.org/

[28] Comet project overview. [Online]. Available: http://www.comet-project.org/overview.html

[29] Fp7convergence project. [Online]. Available: http://www.ict-convergence.eu/

[30] Mobilityfirst future internet architecture project. [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/

[31] (2016) Cyber-secure data and control cloud for power grids. [Online]. Available: http://cdax.eu/

[32] (2016) Greenicn architecture and applications of green information centric networking. [Online]. Available: http://www.greenicn.org/

[33] The ccnx project. [Online]. Available: http://blogs.parc.com/ccnx/

[34] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[35] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[36] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Information centric networking in iot scenarios: The case of a smart home," in *Proc. IEEE Int. Conf. Communications (ICC)*, Jun. 2015, pp. 648–653.

[37] N. Fotiou and G. C. Polyzos, "Realizing the internet of things using information-centric networking," in *2014 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*. IEEE, 2014, pp. 193–194.

[38] F. Wang, L. Hu, J. Zhou, and K. Zhao, "A survey from the perspective of evolutionary process in the internet of things," *International Journal of Distributed Sensor Networks*, vol. 2015, p. 9, 2015.

[39] D. L. Brock, "The electronic product code (epc)," *Auto-ID Center White Paper MIT-AUTOID-WH-002*, 2001.

[40] R. Want, "An introduction to rfid technology," *Pervasive Computing, IEEE*, vol. 5, no. 1, pp. 25–33, 2006.

[41] S. Ahuja and P. Potti, "An introduction to rfid technology," *Communications and Network*, vol. 2, no. 03, pp. 183–186, 2010.

[42] R. CHEN, Y. WANG, Y. LIU, and Z. CHEN, "Rfid anti-collision algorithm based on tags grouping," *Journal of Computer Applications*, vol. 33, no. 8, pp. 2132–2135, 2013.

[43] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[44] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[45] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 553–576, 2016.

[46] J.-P. Vasseur and A. Dunkels, *The 6LoWPAN Adaptation Layer*. Morgan Kaufmann, Boston, 2010, ch. 16.

[47] M. Collotta and G. Pau, "Bluetooth for internet of things: A fuzzy approach to improve power management in smart homes," *Computers & Electrical Engineering*, vol. 44, no. 13, pp. 137–152, 2015.

[48] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, vol. 48, no. 1, pp. 28–35, 2015.

[49] M. Aftanas, "Through wall imaging with uwb radar system," Ph.D. dissertation, TECHNICAL UNIVERSITY OF KOICE Citeseer, 2009.

[50] R. A. Saeed, S. Khatun, B. Ali, and M. Khazani, "Performance of ultra-wideband time-of-arrival estimation enhanced with synchronization scheme," *ECTI Trans. on Electrical Eng., Electronics and Communication*, vol. 4, no. 1, pp. 78–84, 2006.

[51] A. Mrou, M. Heddebaut, F. Elbahhar, A. Rivenq, and J. Rouvaen, "Automatic radar target recognition of objects falling on railway tracks," *Measurement Science and Technology*, vol. 23, no. 2, p. 10, 2012.

[52] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, "Ieee 802.3 az: the road to energy efficient ethernet," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 50–56, 2010.

[53] S. Anbalagan, D. Kumar, G. Raja, W. Ejaz, A. K. Bashir *et al.*, "Sdn-assisted efficient lte-wifi aggregation in next generation iot networks," *Future Generation Computer Systems*, 2017.

[54] Par for 802.11ah. [Online]. Available: http://www.ieee802.org/11/PARs/P802.11ah.pdf

[55] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "Ieee 802.11ah: The wifi approach for m2m communications," *Wireless Communications, IEEE*, vol. 21, no. 6, pp. 144–152, 2014.

[56] Ieee standards association. [Online]. Available: http://standards.ieee.org/news/2014/ieee-802-11ac-ballot.html

[57] A. Attwood, M. Merabti, and O. Abuelmaatti, "Iomanets: Mobility architecture for wireless m2m networks," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 399–404.

[58] S. Hou, M. Wu, W. Liao, and D. Wang, "Performance comparison of aodv and dsr in manet test-bed based on internet of things," in *2015 IEEE 82nd Vehicular Technology Conference (VTC Fall)*. IEEE, 2015, pp. 1–5.

[59] T. A. Ramrekha, O. Adigun, A. Ladas, N. Weerasinghe, and C. Politis, "Towards a scalable routing approach for mobile ad-hoc networks," in *Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2015 IEEE 20th International Workshop on*. IEEE, 2015, pp. 261–266.

[60] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of manet and wsn in iot urban scenarios," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3558–3567, 2013.

[61] R. P. Jover and I. Murynets, "Connection-less communication of iot devices over lte mobile networks," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2015, pp. 247–255.

[62] J. Jermyn, R. P. Jover, I. Murynets, M. Istomin, and S. Stolfo, "Scalability of machine to machine systems and the internet of things on lte mobile networks," in *2015 IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2015, pp. 1–9.

[63] O. Novo, N. Beijar, M. Ocak, J. Kjallman, M. Komu, and T. Kauppinen, "Capillary networks-bridging the cellular and iot worlds," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 571–578.

[64] N. Beijar, O. Novo, J. Jimenez, and J. Melen, "Gateway selection in capillary networks," in *2015 5th International Conference on the Internet of Things (IOT)*. IEEE, 2015, pp. 90–97.

[65] L. D. Closing in on the future with 4g lte and m2m. [Online]. Available: http://www.verizonwireless.com/news/article/2012/09/future-4G-LTE-M2M.html.

[66] G. V. Crosby and F. Vafa, "Wireless sensor networks and LTE-A network convergence," in *Proc. IEEE 38th Conf. Local Computer Networks (LCN)*, Oct. 2013, pp. 731–734.

[67] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: next-generation wireless broadband technology [invited paper]," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 10–22, Jun. 2010.

[68] C. R. Randy. Can the internet of things (iot) survive without satellite? [Online]. Available: http://www.thuraya.com/content/can-internet-things-iot-survive-without-satellite

[69] P. Bhave and P. Fines, "System behavior and improvements for M2M devices using an experimental satellite network," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, May 2015, pp. 13–16.

[70] M. D. Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting internet of remote things," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, Feb. 2016.

[71] M. Ashraf, A. Shahid, J. W. Jang, and K.-G. Lee, "Optimization of the overall success probability of the energy harvesting cognitive wireless sensor networks," *IEEE Access*, vol. 5, pp. 283–294, 2017.

[72] M. A. Shah, S. Zhang, and C. Maple, "Cognitive radio networks for internet of things: Applications, challenges and future," in *Proc. 19th Int Automation and Computing (ICAC) Conf*, Sep. 2013, pp. 1–6.

[73] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, Apr. 2015.

[74] W. Ejaz and M. Ibnkahla, "Machine-to-machine communications in cognitive cellular systems," in *Proc. IEEE Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Oct. 2015, pp. 1–5.

[75] E. Z. Tragos and V. Angelakis, "Cognitive radio inspired M2M communications," in *Proc. 16th Int Wireless Personal Multimedia Communications (WPMC) Symp*, Jun. 2013, pp. 1–5.

[76] T. Hassan, S. Aslam, and J. W. Jang, "Fully automated multi-resolution channels and multithreaded spectrum allocation protocol for iot based sensor nets," *IEEE Access*, 2018.

[77] J. M.-Y. Lim, Y. C. Chang, M. Y. Alias, and J. Loo, "Cognitive radio network in vehicular ad hoc network (vanet): A survey," *Cogent engineering*, vol. 3, no. 1, p. 1191114, 2016.

[78] J. M.-Y. Lim, Y. C. Chang, J. Loo, and M. Y. Alias, "Improving vanet performance with heuristic and adaptive fuzzy logic scheme," *Wireless Personal Communications*, vol. 83, no. 3, pp. 1779–1800, 2015.

[79] S. A. Hussain, M. Iqbal, A. Saeed, I. Raza, H. Raza, A. Ali, A. K. Bashir, and A. Baig, "An efficient channel access scheme for vehicular ad hoc networks," *Mobile Information Systems*, vol. 2017, 2017.

[80] K. M. Alam, M. Saini, and A. E. Saddik, "Toward social internet of vehicles: Concept, architecture, and applications," *IEEE Access*, vol. 3, pp. 343–357, 2015.

[81] M. Doering and L. Wolf, "Opportunistic vehicular networking: Large-scale bus movement traces as base for network analysis," in *Proc. Int High Performance Computing Simulation (HPCS) Conf*, Jul. 2015, pp. 671–678.

[82] Y. Huo, W. Tu, Z. Sheng, and V. C. M. Leung, "A survey of in-vehicle communications: Requirements, solutions and opportunities in iot," in *Proc. IEEE 2nd World Forum Internet of Things (WF-IoT)*, Dec. 2015, pp. 132–137.

[83] M. Naeem, W. Ejaz, L. Karim, S. H. Ahmed, A. Anpalagan, M. Jo, and H. Song, "Distributed gateway selection for m2m communication in cognitive 5g networks," *IEEE Network*, vol. 31, no. 6, pp. 94–100, 2017.

[84] S. B. M. Baskaran, G. Raja, A. K. Bashir, and M. Murata, "Qos-aware frequency-based 4g+ relative authentication model for next generation lte and its dependent public safety networks," *IEEE Access*, 2017.

[85] B. Evans, O. Onireti, T. Spathopoulos, and M. A. Imran, "The role of satellites in 5g," in *Proc. 23rd European Signal Processing Conf. (EUSIPCO)*, Aug. 2015, pp. 2756–2760.

[86] S. Talwar, D. Choudhury, K. Dimou, E. Aryafar, B. Bangerter, and K. Stewart, "Enabling technologies and architectures for 5g wireless," in *Proc. IEEE MTT-S Int. Microwave Symp. (IMS2014)*, Jun. 2014, pp. 1–4.

[87] K. E. Skouby and P. Lynggaard, "Smart home and smart city solutions enabled by 5g, iot, aai and cot services," in *Proc. Int Contemporary Computing and Informatics (IC3I) Conf*, Nov. 2014, pp. 874–878.

[88] C. Boldrini, K. Lee, M. nen, J. Ott, and E. Pagani, "Opportunistic networks," *Computer Communications*, vol. 48, pp. 1–4, jul 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2014.04.007

[89] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, Jan. 2014.

[90] L. M. L. Oliveira, J. Reis, J. J. P. C. Rodrigues, and A. F. de Sousa, "IOt based solution for home power energy monitoring and actuating," in *Proc. IEEE 13th Int. Conf. Industrial Informatics (INDIN)*, Jul. 2015, pp. 988–992.

[91] D. P. F. Möller and H. Vakilzadian, "Ubiquitous networks: Power line communication and internet of things in smart home environments," in *Proc. IEEE Int. Conf. Electro/Information Technology*, Jun. 2014, pp. 596–601.

[92] H. C. Hsieh and C. H. Lai, "Internet of things architecture based on integrated PLC and 3G communication networks," in *Proc. IEEE 17th Int Parallel and Distributed Systems (ICPADS) Conf*, Dec. 2011, pp. 853–856.

[93] G. Bumiller, L. Lampe, and H. Hrasnica, "Power line communication networks for large-scale control and automation systems," *IEEE Communications Magazine*, vol. 48, no. 4, pp. 106–113, Apr. 2010.

[94] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "On the integration of cloud computing and internet of things," in *Proc. Int Future Internet of Things and Cloud (FiCloud) Conf*, Aug. 2014, pp. 23–30.

[95] T. D. P. Bai and S. A. Rabara, "Design and development of integrated, secured and intelligent architecture for internet of things and cloud computing," in *Proc. 3rd Int Future Internet of Things and Cloud (FiCloud) Conf*, Aug. 2015, pp. 817–822.

[96] All the best big data tools and how to use them - import.io. [Online]. Available: https://www.import.io/post/all-the-best-big-data-tools-and-how-to-use-them/.

[97] K. Okoye, U. Naeem, and S. Islam, "Semantic fuzzy mining: Enhancement of process models and event logs analysis from syntactic to conceptual level," *International Journal of Hybrid Intelligent Systems*, no. Preprint, pp. 1–32, 2017.

[98] K. Kotis and A. Katasonov, "Semantic interoperability on the internet of things," *International Journal of Distributed Systems and Technologies*, vol. 4, no. 3, pp. 47–69, 2013. [Online]. Available: http://dx.doi.org/10.4018/jdst.2013070104

[99] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the internet of things," *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, 2012. [Online]. Available: http://dx.doi.org/10.4018/jswis.2012010101

[100] C. C. Aggarwal, N. Ashish, and A. Sheth, "The internet of things: A survey from the data-centric perspective," in *Managing and Mining Sensor Data*. Springer US, dec 2012, pp. 383–428. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-6309-2-12

[101] R. G. Raskin and M. J. Pan, "Knowledge representation in the semantic web for earth and environmental terminology (SWEET)," *Computers & Geosciences*, vol. 31, no. 9, pp. 1119–1125, nov 2005. [Online]. Available: http://dx.doi.org/10.1016/j.cageo.2004.12.004

[102] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the w3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, dec 2012. [Online]. Available: http://dx.doi.org/10.1016/j.websem.2012.05.003

[103] I. K. Ihianle, U. Naeem, S. Islam, and A.-R. Tawil, "A hybrid approach to recognising activities of daily living from object use in the home environment," in *Informatics*, vol. 5, no. 1. Multidisciplinary Digital Publishing Institute, 2018, p. 6.

[104] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks - (HotNets)*. Association for Computing Machinery (ACM), 2011. [Online]. Available: http://dx.doi.org/10.1145/2070562.2070563

[105] I. M. M. Gabriel M. Brito, Pedro Braconnot Velloso, *Information-Centric Networks: A New Paradigm for the Internet*. ISTE LTD, 2013. [Online].

Available: http://www.ebook.de/de/product/19908478/gabriel-m-brito-pedro-braconnot-velloso-igor-m-moraes-information-centric-networks-a-new-paradigm-for-the-internet.html

[106] C. Dannewitz, M. D'Ambrosio, and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks," *Computer Communications*, vol. 36, no. 7, pp. 736–749, apr 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2013.01.014

[107] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. Long, "Managing flash crowds on the internet," in *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on.* IEEE, 2003, pp. 246–249.

[108] M. F. Majeed, S. H. Ahmed, and M. N. Dailey, "Enabling push–based critical data forwarding in vehicular named data networks," *IEEE Communications Letters*, 2016.

[109] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Whlisch, "Information centric networking in the iot: Experiments with ndn in the wild," in *Proceedings of the 1st international conference on Information-centric networking.* Association for Computing Machinery (ACM), 2014, pp. 77–86. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660144

[110] Y. Abidy, B. Saadallahy, A. Lahmadi, and O. Festor, "Named data aggregation in wireless sensor networks," in *Proc. IEEE Network Operations and Management Symp. (NOMS)*, May 2014, pp. 1–8.

[111] M. Amadeo, C. Campolo, and A. Molinaro, "Multi-source data retrieval in IoT via named data networking," in *Proceedings of the 1st international conference on Information-centric networking.* Association for Computing Machinery (ACM), 2014, pp. 67–76. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660148

[112] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *ACM*

SIGCOMM Computer Communication Review, vol. 37, no. 4, pp. 181–192, oct 2007. [Online]. Available: http://dx.doi.org/10.1145/1282427.1282402

[113] The ccnx project. [Online]. Available: http://www.ccnx.org/

[114] S. H. Bouk, S. H. Ahmed, and D. Kim, "Hierarchical and hash based naming with compact trie name management scheme for vehicular content centric networks," *Computer Communications*, vol. 71, pp. 73–83, nov 2015. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2015.09.014

[115] S. Li, Y. Zhang, D. Raychaudhuri, and R. Ravindran, "A comparative study of mobilityfirst and ndn based icn-iot architectures," in *Proc. 10th Int Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine) Conf*, Aug. 2014, pp. 158–163.

[116] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang, "Securing building management systems using named data networking," *IEEE Network*, vol. 28, no. 3, pp. 50–56, 2014.

[117] Psirp project. [Online]. Available: http://www.psirp.org/

[118] Fp7-sail project. [Online]. Available: http://www.sail-project.eu/

[119] Fp7-4ward project. [Online]. Available: http://www.4ward-project.eu/.

[120] P. P. Ray, M. Mukherjee, and L. Shu, "Internet of things for disaster management: State-of-the-art and prospects," *IEEE Access*, vol. 5, pp. 18 818–18 835, 2017.

[121] M. Amadeo, C. Campolo, and A. Molinaro, "Internet of things via named data networking: The support of push traffic," in *Proc. Int Network of the Future (NOF) Conf. and Workshop the*, Dec. 2014, pp. 1–5.

[122] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang, "Securing building management systems using named data networking," *IEEE Network*, vol. 28, no. 3, pp. 50–56, May 2014.

[123] S. H. Bouk, S. H. Ahmed, D. Kim, and H. Song, "Named-data-networking-based its for smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 105–111, 2017.

[124] D. Evans. (2011) The internet of things how the next evolution of the internet is changing everything. [Online]. Available: http://www.cisco.com/c/dam/en-us/about/ac79/docs/innov/IoT-IBSG-0411FINAL.pdf

[125] S. Condon. (2016) Iot will account for nearly half of connected devices by 2020, cisco says, the number of machine-to-machine connections should grow from 4.9 billion in 2015 to 12.2 billion in 2020, according to cisco's annual visual networking index. [Online]. Available: http://www.zdnet.com/article/iot-will-account-for-nearly-half-of-connected-devices-by-2020-cisco-says/

[126] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: A survey and future perspectives," *Communications Magazine, IEEE*, vol. 54, no. 2, pp. 98–104, 2016.

[127] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, Mar. 2016.

[128] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[129] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[130] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2004, pp. 918–928.

[131] H. Zhang, W. Quan, J. Guan, C. Xu, and F. Song, "Uniform information with a hybrid naming (hn) scheme, draft-zhang-icnrg-hn-04. txt," ICNRG Internet Draft, Expires, 7 Apr, Tech. Rep., 2016.

[132] ——, "Uniform information with a hybrid naming (hn) scheme, draft-zhang-icnrg-hn-04. txt," ICNRG Internet Draft, Expires, 7 Oct, Tech. Rep., 2016.

[133] J. Burke, P. Gasti, N. Nathan, and G. Tsudik, "Securing instrumented environments over content-centric networking: the case of lighting control and ndn," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 2013, pp. 394–398.

[134] ——, "Secure sensing over named data networking," in *Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on*. IEEE, 2014, pp. 175–180.

[135] S. H. Bouk, S. H. Ahmed, and D. Kim, "Hierarchical and hash-based naming scheme for vehicular information centric networks," in *Proc. Int. Conf. Connected Vehicles and Expo (ICCVE)*, Nov. 2014, pp. 765–766.

[136] B. Saadallah, A. Lahmadi, and O. Festor, "Ccnx for contiki: implementation details," Ph.D. dissertation, INRIA, 2012.

[137] N.-T. Dinh and Y. Kim, "Potential of information-centric wireless sensor and actor networking," in *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*. IEEE, 2013, pp. 163–168.

[138] J. Hong, W. Chun, and H. Jung, "A flat name based routing scheme for information-centric networking," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2015, pp. 444–447.

[139] B. Li, A. P. Verleker, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for icn naming scheme," in *2014 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2014, pp. 391–399.

[140] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Social cooperation for information-centric multimedia streaming in highway vanets," in *World of*

Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a. IEEE, 2014, pp. 1–6.

[141] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed, and S. Aslam, "Hierarchical and flat based hybrid naming scheme in content-centric networks of things," *IEEE Internet of Things*, vol. 5, no. 2, pp. 1070–1080, 2018.

[142] A. Dokic. (2007) Micaz and telosb sensor device driver port to contiki. [Online]. Available: http://www.eecs.jacobs-university.de/archive/bsc-2007/djokic.pdf

[143] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–53, 2012.

[144] S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "Comparison of naming schema in icn," in *The 22nd IEEE International Symposium on Local and Metropolitan Area Networks(LANMAN)*. IEEE, 2016.

[145] Y. Sun, Y. Zhang, H. Zhang, B. Fang, and X. Du, "Geometric routing on flat names for icn," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[146] Y. Sun, Y. Zhang, S. Su, H. Zhang, and B. Fang, "Geometric name routing for icn in dynamic world," *China Communications*, vol. 12, no. 7, pp. 47–59, 2015.

[147] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in icn: Attribute-based encryption and routing," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM, 2013, pp. 39–40.

[148] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Scalable name lookup with adaptive prefix bloom filter for named data networking," *IEEE Communications Letters*, vol. 18, no. 1, pp. 102–105, 2014.

[149] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, "Beacon synchronization and duty-cycling in ieee 802.15. 4 cluster-tree networks: A review," *IEEE Internet of Things Journal*, 2018.

[150] K. Williamson, *Research methods for students, academics and professionals: Information management and systems.* Elsevier, 2002.

[151] ——, *Research methods for students, academics and professionals: Information management and systems.* Elsevier, 2002.

[152] D. Evans, "The internet of things how the next evolution of the internet is changing everything (april 2011)," *White Paper by Cisco Internet Business Solutions Group (IBSG)*, 2012.

[153] G. N. Cristina, G. V. Gheorghita, and U. Ioan, "Gradual development of an iot architecture for real-world things," in *Modelling Symposium (EMS), 2015 IEEE European.* IEEE, 2015, pp. 344–349.

[154] S. JOSE. (2017) Cisco visual networking index predicts near-tripling of ip traffic by 2020. [Online]. Available: https://newsroom.cisco.com/press-release-content?type=press-release&articleId=1771211

[155] S. H. Ahmed, S. H. Bouk, and D. Kim, *Content-centric networks: an overview, applications and research challenges.* Springer, 2016.

[156] Y. Zhang, D. Raychadhuri, L. Grieco, E. Baccelli, J. Burke, and G. Wang, "Icn based architecture for iot - requirements and challenges draft-zhang-iot-icn-challenges-02," *ICN Research Group, Internet-Draft*, 2016. [Online]. Available: https://tools.ietf.org/html/draft-zhang-iot-icn-challenges-02-page-12

[157] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang, "Named data networking of things," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI).* IEEE, 2016, pp. 117–128.

[158] M. A. M. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "On the performance of caching and forwarding in information-centric networking for the iot," *Wired/Wireless Internet Communications*, pp. 313–326, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-22572-2-23

[159] S. H. Bouk, S. H. Ahmed, and D. Kim, "Ndn goes deep: foreseeing the underwater named data networks," in *Proceedings of the Symposium on Applied Computing*. ACM, 2017, pp. 642–646.

[160] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*. ACM, 2010, pp. 3–8.

[161] G. Fowler, L. C. Noll, K.-P. Vo, and D. Eastlake, "The fnv non-cryptographic hash algorithm," *Ietf-draft*, 2011.

[162] (2016) Fowler noll vo hash function. [Online]. Available: http://will.thimbleby.net/algorithms/doku.php?id=fowler-noll-vo-hash-function

[163] H. Lamaazi, N. Benamar, M. I. Imaduddin, and A. J. Jara, "Performance assessment of the routing protocol for low power and lossy networks," in *Wireless Networks and Mobile Communications (WINCOM), 2015 International Conference on*. IEEE, 2015, pp. 1–8.

[164] S. Jabbar, M. Khan, B. N. Silva, and K. Han, "A rest-based industrial web of things framework for smart warehousing," *The Journal of Supercomputing*, pp. 1–15, 2016.

[165] A. Ahmad, A. Paul, M. M. Rathore, and H. Chang, "Smart cyber society: Integration of capillary devices with high usability based on cyber–physical system," *Future Generation Computer Systems*, vol. 56, pp. 493–503, 2016.

[166] A. Ahmad, A. Paul, M. Rathore, and H. Chang, "An efficient multidimensional big data fusion approach in machine-to-machine communication," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 15, no. 2, p. 39, 2016.

[167] Y. Agarwal and A. K. Dey, "Toward building a safe, secure, and easy-to-use internet of things infrastructure," *Computer*, vol. 49, no. 4, pp. 88–91, 2016.

[168] K. Xu, X. Wang, W. Wei, H. Song, and B. Mao, "Toward software defined smart home," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 116–122, 2016.

[169] (2016) Smart campus solution for education. [Online]. Available: http://e.huawei.com/ae/solutions/industries/education/higher-education/smart-campus

[170] R. Haq. (2016, September 12,) The express tribune ¿ pakistan 13 varsities to be turned into smart campuses. [Online]. Available: https://tribune.com.pk/story/1181541/wi-fi-access-13-varsities-turned-smart-campuses/

[171] G. C. Lazaroiu, V. Dumbrava, M. Costoiu, M. Teliceanu, and M. Roscia, "Energy-informatic-centric smart campus," in *Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on.* IEEE, 2016, pp. 1–5.

[172] A. Adamkó, T. Kãdek, and M. Kósa, "Intelligent and adaptive services for a smart campus," in *Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on.* IEEE, 2014, pp. 505–509.

[173] M. Guo and Y. Zhang, "The research of smart campus based on internet of things & cloud computing," in *Wireless Communications, Networking and Mobile Computing (WiCOM 2015), 11th International Conference on.* IET, 2015, pp. 1–6.

[174] D. Van Merode, G. Tabunshchyk, K. Patrakhalko, and G. Yuriy, "Flexible technologies for smart campus," in *Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on.* IEEE, 2016, pp. 64–68.

[175] S. H. Ahmed and D. Kim, "Named data networking-based smart home," *ICT Express*, vol. 2, no. 3, pp. 130–134, 2016.

[176] U. De Silva, A. Lertsinsrubtavee, A. Sathiaseelan, and K. Kanchanasut, "Named data networking based smart home lighting," in *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, 2016, pp. 573–574.

[177] U. De Silva, A. Lertsinsrubtavee, A. Sathiaseelan, C. Molina-Jimenez, and K. Kanchanasut, "Implementation and evaluation of an information centric-based smart lighting controller," in *Proceedings of the 12th Asian Internet Engineering Conference*. ACM, 2016, pp. 1–8.

[178] 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 *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2017, pp. 289–301.

[179] (2012) 50 sensor applications for a smarter world. [Online]. Available: http://www.libelium.com/50-sensor-applications/

[180] D. Vigano. (2017) Transforming iot into iome (internet of me) via smart garments. [Online]. Available: http://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Transforming-IoT-into-IoMe-internet-of-me-via-smart-garments

[181] A. B. Colombo. (2016) Smart buildings get smarter with iot. [Online]. Available: https://facilityexecutive.com/2016/12/smart-buildings-get-smarter-with-iot/

[182] S. Arshad, M. A. Azam, S. H. Ahmed, and J. Loo, "Towards information-centric networking (icn) naming for internet of things (iot): The case of smart campus," in *Proceedings of the International Conference on Future Networks and Distributed Systems*, ser. ICFNDS '17. New York, NY, USA: ACM, 2017, pp. 41:1–41:6. [Online]. Available: http://doi.acm.org/10.1145/3102304.3102345

[183] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent advances in information-centric networking based internet of things (icn-iot)," *IEEE Internet of Things (In Press)*, 2018.

[184] M. F. Majeed, S. H. Ahmed, and M. N. Dailey, "Enabling push-based critical data forwarding in vehicular named data networks," *IEEE Communications Letters*, vol. 21, no. 4, pp. 873–876, 2017.

[185] A. Compagno, M. Conti, and R. E. Droms, "Onboardicng: a secure protocol for on-boarding iot devices in icn." in *ICN*, 2016, pp. 166–175.