Recent Advances in Information-Centric Networking based Internet of Things (ICN-IoT)

Sobia Arshad, Muhammad Awais Azam, Mubashir Husain Rehmani, Jonathan Loo

***Abstract*—Information-Centric Networking (ICN) is being realized as a promising approach to accomplish the shortcomings of current IP-address based networking. ICN models are based on naming the content to get rid of address-space scarcity, accessing the content via name-based-routing, caching the content at intermediate nodes to provide reliable, efficient data delivery and self-certifying contents to ensure better security. Obvious benefits of ICN in terms of fast and efficient data delivery and improved reliability raises ICN as highly promising networking model for Internet of Things (IoTs) like environments. IoT aims to connect anyone and/or anything at any time by any path on any place. From last decade, IoTs attracts both industry and research communities. IoTs is an emerging research field and still in its infancy. Thus, this paper presents the promise of ICN for IoTs by providing state-of-the-art literatures. We discuss briefly the feasibility of ICN features and their models (and architectures) in the context of IoT. Subsequently, we present a comprehensive survey on ICN based caching, naming, security and mobility approaches for IoTs with appropriate classification. Furthermore, we present operating systems (OS) and simulation tools for ICN-IoT. Finally, we provide important research challenges and issues faced by ICN for IoTs.**

***Index Terms*—IoT, ICN, NDN, CCN, Information-Centric Networking, ICN-IoT Caching Schemes, ICN-IoT Naming Schemes, ICN-IoT Security Schemes, ICN-IoT Mobility Schemes, Taxonomy.**

I. Introduction

A. Motivation and 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, smart-phones, 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 of these devices connect to the Internet, generation of large amount of data is an apparent consequence. Moreover, this IoT data has to combine with the data produced from Facebook likes and Youtube videos which results in IoT Big Data. Therefore, efficient access and discovery of IoT Big Data put more constraints on the underlying TCP/IP architecture while raising many important issues.

Among these issues (from IoT device perspective), one is naming and addressing every IoT device [2]-[3]. As IPv4 addressing space is exhausted, IPv6 address space may also exhaust in the future. Besides this, IPv6 address is quite long and its long length makes it less suitable for communication through 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. Furthermore, every device has different constraints and specifications which raise another issue of heterogeneity. This is due to the fact that IoTs comprises on devices which are heterogeneous in terms of processing power capability, size, memory, battery life and cost. While 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. Besides heterogeneity, in these low memory and low battery life constraint-oriented devices, data can become unavailable most of the time which causes data unavailibility. Therefore, solutions like in-network caching (which are reuired to make data available) are missing in naive IP-based networking. In addition, 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 require better mobility handling [8]-[9].

On the other hand, from data perspective, most of the IoTs application users are more interested in getting the updated information rather than knowing the address of information source. As an instance, IoT devices especially in the domain called wireless sensor networks (WSN), have specific purpose to harvest information 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 value of a certain area rather than the temperature value from a specific sensor.

Considering TCP/IP as network architecture for IoTs, which was traditionally designed to connect limited number of computers and to share limited and expensive network resources through limited address space at network layer, it is definitely not designed to fulfill IoTs requirements. Moreover, besides above-mentioned requirements, IoTs huge data put additional requirements like data dissemination and scalability on the underlying architecture. To fulfill all these needs of IoTs, Information-Centric Networking (ICN) (which is a promising

Sobia Arshad and Muhammad Awais Azam are with the Department of Computer Engineering, University of Engineering & Technology, Taxila, Pakistan. (email(s):sobia.arshad@uettaxila.edu.pk, awais.azam@uettaxila.edu.pk) Phone:+92 (0)331 9811899

Mubashir Husain Rehmani is with the Waterford Institute of Technology, Ireland.(email: mshrehmani@gmail.com) Phone:+92 (0)333 3052764)

Jonathan Loo is with the University of West London, London, UK. (jonathan.loo@uwl.ac.uk)

Table I
LIST OF ACRONYMS USED

Acronyms	Definitions	Acronyms	Definitions
6LowPANs	*IPv6 over Low power Wireless Personal Area Networks*	**CCN**	*Content-Centric Networking*
CS	*Content Store*	**COMET**	*COntent Mediator architecture for content aware nETworks*
CONET	*Content Network*	**DF**	*Destination Flag*
DONA	*Data Oriented Network Architecture*	**DoS attack**	*Denial-of-Service attack*
DPI	*Deep Packet Inspection*	**FIA**	*Future Internet Architecture*
FIA-NP	*FIA-Next Phase*	**FIB**	*Forwarding Information Base*
FP7	*Framework Programme 7*	**GPRS**	*General Packet Radio Service*
GSM	*Global System for Mobile communication*	**GUID**	*Globally Unique Identifier*
IERC	*IoT European Research Cluster*	**ICN**	*Information Centric Networking*
IoT	*Internet of Things*	**IPV4**	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*	**LRU**	*Least Recently Used*
LTE	*Long Term Evolution*	**LTE-A**	*LTE Advanced*
M2M	*Machine-to-Machine*	**MF**	*MobilityFirst*
NDN	*Named Data Networking*	**NetInf**	*Networking of Information*
NFC	*Near Field Communication*	**NRS**	*Name Resolution System*
NSF	*National Science Foundation*	**PARC**	*Palo Alto Research Center*
PIT	*Pending Interest Table*	**PSIRP**	*Publish-Subscribe Internet Routing Paradigm*
PURSUIT	*Publish SUbscribe Internet Technology*	**SAIL**	*Scalable and Adaptive Internet soLutions*
SIT	*Satisfied Interest Table*	**TCP/IP**	*Transmission Control Protocol/Internet Protocol*

candidate for the future Internet foundation) has recently emerged as an ideal candidate. So far, there are nine major architectures proposed under the concept of ICN including DONA, CCN [11], PURSUIT [12], NetInf [13], CURLING [14], CONET [15], MobilityFirst [16], C-DAX [17] and Green ICN [18]. 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 [19]. ICN primary characteristics include in-network caching, naming the contents, better and easy mobility management, improved security and scalable information delivery which are naturally suitable for IoT applications. Moreover, ICN-based hourglass architecture provides us thin-waist like TCP/IP [20]. Additionally, ICN can mask over TCP/IP network layer or MAC layer. CCN could be applied just above MAC layer especially in WSN. Current literature [21]-[22] argue that ICN seems to replace IP, rather we believe and foresee ICN is an overlay network sitting on IP network. In fact, CCN is a layer that mask the need of associating content with the IP address instead by name. The actual content delivery still require TCP/IP interface or direct MAC (layer 2) interface.

ICN's striking feature in-network caching, can efficiently handle 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 [23]-[21]. That's why in this article we survey ICN-based naming, in-network caching, security and mobility schemes which are explored for IoTs. List of acronyms used in this paper is provided in Table I.

B. Review of Related Survey Articles

Our current survey on ICN-based IoTs is unique from the prior surveys as we survey holistically ICN-based IoTs caching, ICN-based IoTs naming, ICN-based IoTs security and ICN-based IoTs mobility schemes. A plenty of surveys is available on either alone IoTs or on specifically ICN related issues. To the best of our knowledge, this work is the only detailed survey that emphasizes ICN for IoTs.

Exclusively IoT emphasized surveys have covered the IoT basics including building blocks and characteristics, enabling technologies, smart potential applications, projects and related research challenges in [2], [5], [10]. Eight research directions for IoTs are listed down in [6]. Context awareness solutions for IoTs are discussed in [27]. Middle-ware requirements and solutions are surveyed in [24]. IoTs security issues and their corresponding solutions are outlined in [25]. In [28], specifically Sybil attacks in IoTs are discussed along with their defense schemes. Moreover, classification of Operating Systems (OSs) for IoTs is presented in [26]. List of survey paper for IoTs is provided in Table II.

Surveys that solely focused ICN include [20], in which general ICN is described along with four ICN architectures including DONA, CCN, PSIRP and NetInf. George Xylomenos et al., in [21] described ICN concept, its features and extended the research of [20] by adding three more updated architectures named CONVERGENCE, CONET and MobilityFirst. Moreover, [32] focused on ICN energy efficient caching schemes on the basis of content placement, cache placement and

Table II
IOTS AND ICN RELATED SURVEY ARTICLES

IoTs Related Survey Articles			
Sr#	**Reference(s)**	**Topics Covered**	**Publication Year**
1.	[24]	IoT middleware requirements and solutions	2016
2.	[25]	IoT security Issues and their corresponding solutions	2015
3.	[26]	Classification of IoT Oss	2015
4.	[6]	Eight research directions for IoTs	2014
5.	[27]	Context awareness solutions for IoT	2014
6.	[28]	Sybil attacks in IoTs have been discussed along with defense schemes	2014
7.	[10]	Basics of IoT including building blocks and characteristics of IoTs, IoT enabling technologies, smart potential applications, projects and related research challenges	2015
8.	[5]		2014
9.	[2]		2010
ICN Related Survey Articles			
Sr#	**Reference(s)**	**Topics Covered**	**Publication Year**
1.	[29]	ICN for VANETs and Future Directions	2016
2.	[30]	Taxonomy of security attacks and naming corresponding solutions	2015
3.	[31]	caching mechanisms,performance parameters	2015
4.	[32]	ICN energy efficient caching schemes, content placement,cache placement and request-to-cache routing	2014
5.	[21]	Seven ICN Architectures and Research Directions	2014
6.	[33]	Routing and naming schemes	2012
7.	[20]	Four ICN Architectures	2012
ICN for IoT Survey Article			
Sr#	**Reference**	**Topics Covered**	**Publication Year**
1.	[34]	Briefly identify ICN for IoT and Future Directions	2016

request-to-cache routing. While [31] discussed only NDN and DONA architectures, summarized caching mechanisms, described performance parameters and conducted simulations for the evaluation of caching mechanisms. Routing and naming schemes for ICN are covered in [33]. Comprehensive survey of possible attacks in ICN is presented in [30]. Moreover, taxonomy of security attacks (i.e. categorized into naming, caching, routing and other attacks) in ICN is presented and their existing solutions are discussed. ICN for VANETs along with future research directions is presented in [29]. ICN related literature is listed in Table II.

One pioneer short article [34] that identifies ICN for IoT, surveys briefly ICN for IoT without providing enough literature survey. In contrast to [34], our present survey, provides comprehensive up-to-date review of ICN for IoT, including ICN models and their feasibility for IoT, additionally caching techniques, naming schemes, security schemes and mobility handling mechanisms along with operating systems, simulators and detail research challenges for ICN-IoT research community.

C. Contribution of This Survey Article

We mainly aim to discuss ICN for IoTs. To meet our aim we provide holistic and comprehensive literature on ICN-based in-network caching, ICN content naming schemes, ICN security schemes and ICN mobility handling schemes for IoTs. With such goals, to the best of our knowledge, it makes this paper pioneer and unique in this field. We outline the details of contributions we made as:

- We provide very brief overview of IoT architecture requirements and major ICN architectures w.r.t their suitability for IoTs in terms of naming, caching, security and mobility handling schemes.
- We summarize ICN-based architectures for IoT.
- We provide comprehensive survey of ICN-based in-network caching techniques for IoTs and classification of these schemes on the basis of role of content and node properties in ICN caching mechanisms for IoT.
- We provide classification of ICN-based content naming approaches on the basis of name structures for IoTs.
- We classify ICN-based security schemes for IoTs on the basis of their security handling for IoT contents and IoT devices.
- We categorize ICN-based mobility schemes into IoT producer mobility and hand-off management.
- We classify famous ICN-IoT simulators and OSs and identify ndnSIM as a more explored tool for ICN-IoT.
- We provide issues, challenges and future research directions which ICN is facing for IoTs.

D. Organization of the paper

The rest of the paper is organized as follows. Section II provides a brief overview about IoT network architecture requirements, ICN models feasibility for IoT with respect to their

naming, caching, security and mobility handling mechanisms and ICN-based architectures for IoTs. In sections III, IV, V, VI, ICN-based caching techniques, naming approaches, security and mobility support are discussed, respectively. Section VII presents available OSs and simulators for ICN-IoTs. In section VIII, we present open challenges and future trends of ICN into IoT. Finally, section IX concludes the paper.

II. Information-Centric Networking (ICN) Suitability for IoTs

As IoTs is the connectivity of things through the unified Internet. Things can be humans and smart machines of any sort and this is illustrated in the lower portion of Fig. 1. These things can connect in three ways (connectivity in IoTs can be seen in upper portion of Fig. 1): i) Machine-type-Communication (MTC), ii) Machine-to-Human (M2H) and iii) Human-to-Human (H2H). IoT works in four major steps namely: i) Data acquisition or data sensing, ii) Data transmission, iii) Data Processing and Information management and iv) Action & Utilization. These major IoT working phases and corresponding elements can be visualized in Fig. 2 and related literature is listed in Table. III.

This section fulfills six purposes: Firstly, we list and describe IoTs architecture requirements. However, our aim is not to survey and discuss IoTs in depth rather we illustrate it to highlight the related issues and identify architecture requirements. Secondly, we discuss IP-based evolutionary approaches for IoTs. Thirdly, we present the limitations of IP-based approaches. Fourthly, we provide mapping of IoT requirements against ICN characteristics. In next sub-section, we describe briefly ICN-based proposed architectures w.r.t their naming, caching, security and mobility feasibility for IoTs and lastly, we present some approaches which discuss and explore ICN for IoTs.

A. IoTs Architecture Requirements

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

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 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) 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.

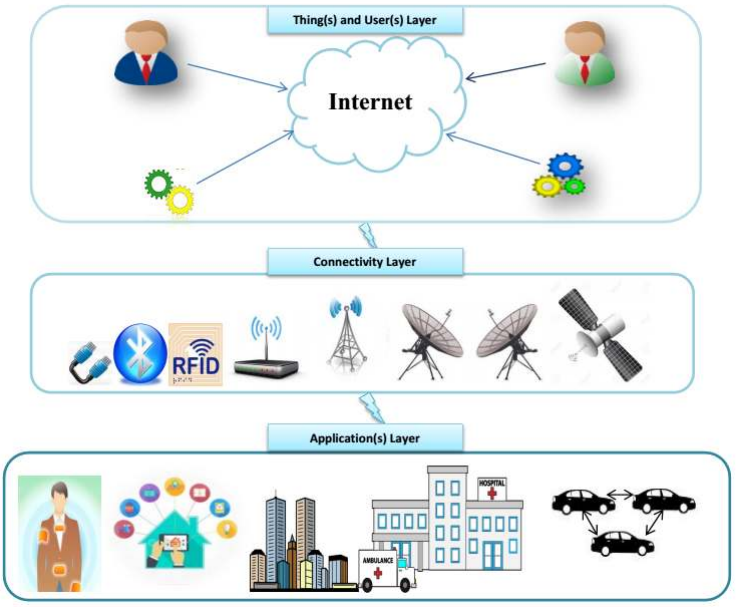

Figure 1. Internet of Things (IoTs): Connectivity Types, Internet Technologies and IoTs Smart Applications.

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 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.

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. 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.

5) Heterogeneity and Interoperability: As we have seen above that RFID tags and smart sensors mainly build IoTs.

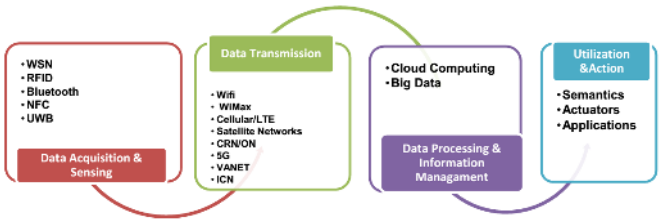

Figure 2. Phases in IoT and Corresponding Enabling Technologies

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 inter-operable way.

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.

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.

B. Evolutionary TCP/IP Approaches for IoTs

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 [35]-[36]-[37].

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)[38] group designed a framework for smart applications to work efficiently on IPv6-based constraint-oriented smart devices. Constrained Application Protocol (CoAP)[39] 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.

Table III
IoTs Phases and Corresponding Technologies

IoT Phase		Components and Reference(s)
Acquisition and Sensing		RFID[47] WSN [27], [24] Bluetooth[48] NFC[49] UWB [50]
Data Transmission	Current Enabling Technologies	Ethernet[51] Wi-Fi[52], [53] Wi-MAX MANETs[54] Cellular Networks[55], [56], [57] Satellite Networks [58]
	Future Enabling (or Enabled by IoTs) Technologies	CRN[59] VANETs [60] 5G [61] ON[62] PLC[63]
Data Processing and Info. Management		Cloud Computing[64] Big Data[65]
Action and Utilization		Semantics[10], [66] Actuators[10] Applications[1], [10]

CoAP runs over UDP to provide better communication among resource-oriented devices. IPv6 over Low Power Wireless Personal Area Networks working group (6LoWPAN-WG) [40] 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)[41] 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 network. 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) [42] 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 [43] is focusing to build minimal and inter operable IP protocol stack for constraint-oriented IoT devices. And the Thing-2-Thing Research Group (T2TRG)[44] 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)[45] 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 [46]. Nonetheless, above mentioned projects for IoT architecture lies under 'all-IP architectures' umbrella.

And IP-based networking is inherently designed for host-

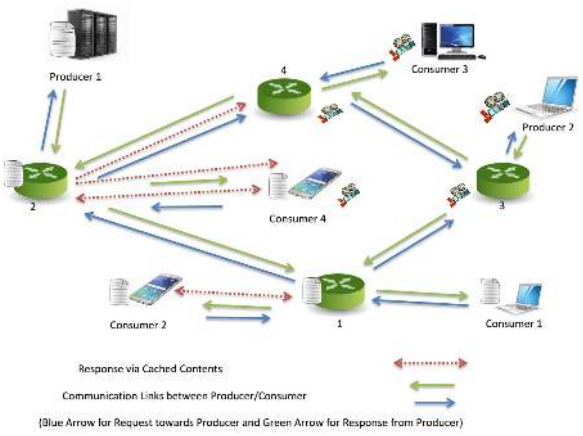

Figure 3. ICN Operation: Consumer Requests for a Specific Content by Nearest Routers (1,2,3,4) and Producer Replies and Intermediate Nodes Caches that Content and Fulfills Further Request through Cached Contents Rather Than Sending Request Towards the Original Producer

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 [67]. Due to this, IETF has also started ICN research group that will help to evolve IP-based architecture [68].

C. Limitations of TCP/IP Architecture and Importance of ICN for IoTs

From both, today's Internet and IoT context, as 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. In addition, flash crowds are also the obvious consequence of todays Internet usage [20], [21], [69], [70], [71]. 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) [72] and data can become unavailable due to end of batteries of many sensors located in that producer vehicle. 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. With the help of ICN in-network caching, intermediate routers (any vehicle) can provide data on behalf of original producer who cached

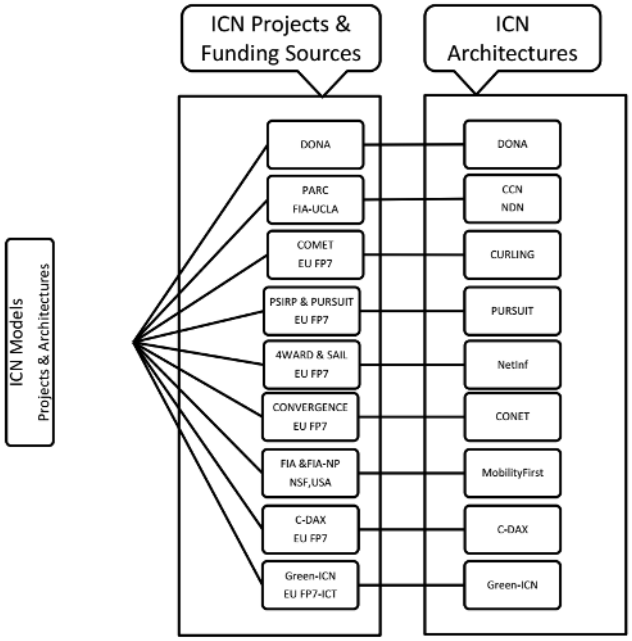

Figure 4. ICN Projects, Funding Sources and Architectures

that information item while reducing so-called flash crowd situation. As ICN offers in-network caching which makes it more ideal for low power devices. 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 [73]. Furthermore data can only be accessed whenever receiver explicitly requests a data. As data is searched on the basis of its location-independent name. This provides *opaque communication* between sender and receiver making it more secure. Details of ICN (specifically NDN) operation is shown in Fig. 3.

D. IoT Requirements Mapping to ICN Characteristics

IoT applications which need scalability in terms of support for 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 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

Table IV
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

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 IV summarizes the mapping of IoT requirements to supporting ICN features.

E. Feasibility of ICN Models and Projects for IoTs

This sub-section presents naming, caching, security and mobility support of nine famous ICN architectures such as DONA [74], NDN, COMET, PURSUIT, SAIL, CONVERGENCE, MobilityFirst, C-DAX [17] and Green ICN [18]. ICN major projects and architectures along with funding sources are presented in Fig. 4 and their feasibility w.r.t naming, caching, security and mobility support is summarized in Table V. However, further details of these architectures can be found in [21].

F. ICN-based IoT Architectures

In this sub-section, we present ICN-based IoT research efforts (in following paragraphs) 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 [67]. 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. 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 (NDN) based architecture. To support IoT **push operations**, three different strategy schemes are presented to provide push-type communication for NDN in [75]. 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 [76]. 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)[77]. 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

Table V
ICN PROJECTS, CORRESPONDING ARCHITECTURES AND THEIR FEASIBILITY FOR IOT

Project Name, Duration and Funding Source	ICN Architecture Name	1. Naming, 2. Caching, 3. Security and 4. Mobility	Extent of Suitability for IoT Applications
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	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 (Sep 2010-Feb 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
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	MobilityFirst 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.
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

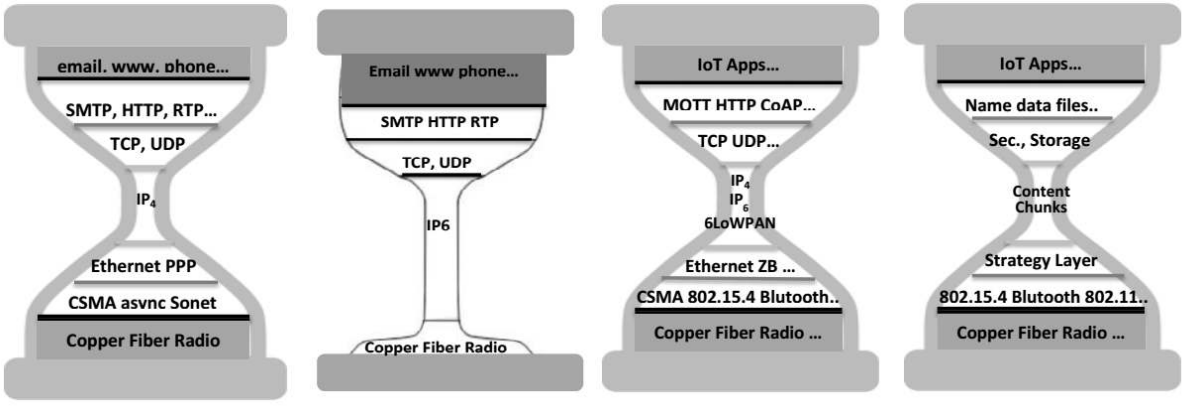

Figure 5. IP-based Network Architectures and ICN-based IoT Network Architecture

[78]. 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 following four sections, we categorize and present ICN-based IoT research through ICN caching, naming, security and mobility support which is explored for IoT environment.

III. ICN-IoT Caching Schemes

Inherently, the current Internet is designed to forward all requests of same content towards original producer and hence increases network load, retrieval delay and bandwidth consumption. The current Internet lacks support for data dissemination and fast retrieval of the content. These issues raised the need of in-network caching. To cope these shortcomings of the current Internet architecture, *Content Delivery Networks* (CDNs) were introduced. By employing CDNs, caching is deployed as an overlay patch at application layer (web-caching) of the current Internet architecture. CDNs are costly to implement and do not utilize network resources efficiently in case of dynamic flash crowds. Thus, in the design of the future Internet architecture caching is added as an important feature. In ICN-based future Internet architectures, caching is implemented at network layer that directly operates on named information. ICN architectures DONA, NDN, SAIL and MobilityFirst primarily support on-path caching while PURSUIT, COMET and CONVERGENCE support both on-path and off-path caching [21].

In ICN-based IoT, caching is highly required to disseminate information quickly towards edge devices in a cost-efficient way. As some IoT applications need fresh contents with some specific timing requirements. And mostly, IoT contents are ephemeral in nature that need to replace with the newer versions, for instance, temperature value of a room needs to be monitored and updated continuously. Moreover, as IoT nodes are highly heterogeneous that may differ in the processing resources (i.e., constraint-oriented and powerful nodes) and IoT networks are mixture of wired and wireless technologies.

In IoTs, caching at intermediate devices or routers offers many benefits. As receiver is dissociated from original producer, therefore by caching the contents, security improves and scalability of IoTs network increases [67]. Energy efficiency of contraint oriented devices can be improved and mobility can be handled in more better ways [34]. Resiliency and life of IoT networks can be improved by employing caching carefully [79].

As caching offer many advantages, it also puts same restrictions and complications on the design of caching strategies for environment like IoTs. To design ICN-based caching for IoTs, caching strategies must count for some properties of content to cache and node that intends to cache it. Content properties can include popularity, freshness, ephemerality, timing and specific producer while caching node properties can count for battery (power level), distance of node from producer (or/and consumer) and remaining memory. On the basis of this mentioned observation, we provide caching placement strategies into following three categories:

1) Content-Based Caching (CBC), these strategies decide what content to store on the basis of content properties.

2) Content and Node-Based Caching (CNBC), these schemes decide whether a node should cache content or not, depending on both content properties and node resources (like battery life).

3) Alternative Caching Schemes, algorithms that include distance of a node from producer or position/role in network in caching decision lies in this category. ICN-based caching node architecture and cache coherency are also discussed in this sub-section.

An overview of ICN-based caching schemes for IoTs is presented and summarized in Table VI. A caching strategy is further divided into following three phases:

1) Content placement into cache, in this phase cache space is allocated to contents on the basis of content and/or node. Content placement schemes include cache each and everything (universal caching), probabilistic caching etc.

2) Content replacement from cache, in this second phase, when cache becomes full with contents and there is no space vacant for next upcoming content, it is decided to which already existing content it will replace. Content replacement schemes include LRU (Least Recently Used), LFU (Least Frequently Used) etc.

3) Cache coherency of contents in cache, in this phase, validity of contents residing in cache is checked.

Caching performance measures include retrieval delay, hit ratio, network lifetime (how long network will exist in terms of connectivity), interest re-transmissions (total number of interest sent to get a content) and energy consumption per content (how much energy is required to decide about cache a content and/or replace it). ICN-based caching placement methods have been extensively investigated in the context of IoTs in [80], [81], [82], [83], [84] as depicted in Fig. 6. In the following subsections, we survey caching placement schemes along with caching replacement schemes. According to Fig. 6, we sub-classify caching placement schemes into three categories: Content-Based Caching (CBC), Content and Node-Based Caching (CNBC) and alternative caching schemes.

We further classify CBC on the basis of freshness, probability and CNBC schemes according to freshness, popularity along with node properties. We sub-classify alternative caching schemes into infrastructure-based caching, caching node architecture and cache coherency.

A. Content-Based Caching (CBC) for ICN-IoT

Most of IoT applications that process the contents put rigorous constraints on the contents. Some IoT applications demand contents with freshness constraints while other may demand the content with high probability. Probability for a content, can be set according to the popularity or in a random fashion. In this section, we present ICN-based caching strategies for IoTs, those include such content properties in caching decision.

1) Freshness of Content: IoT contents required by IoT applications are transient in nature that update their values continuously (e.g. temperature sensors update their values and consumer could request the most recent value or of specific date or time). Updated information can be received through specifying freshness value. Thus, caching strategies dealing with freshness are highly important for ICN-based IoTs. In

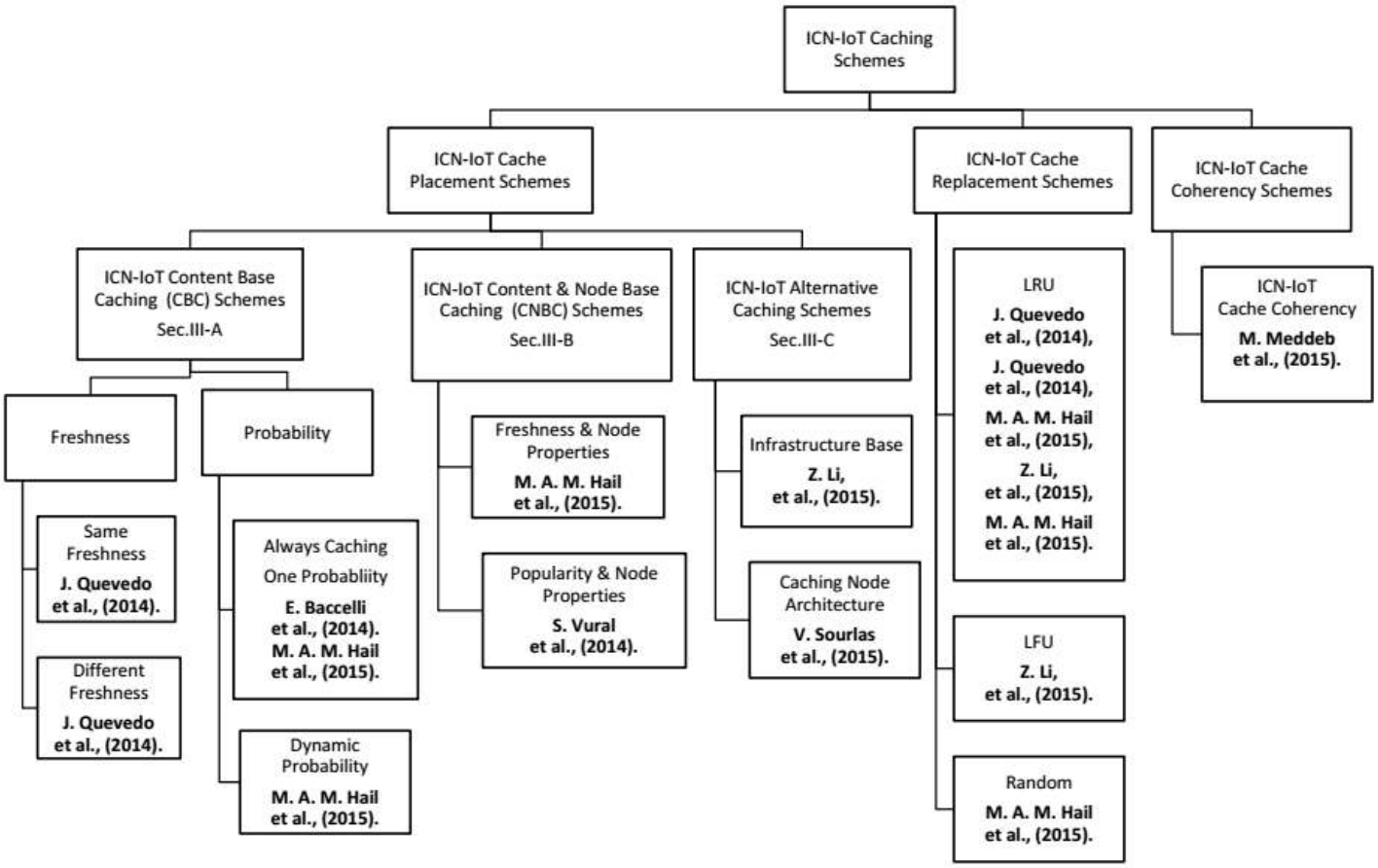

Figure 6. ICN-IoT In-Network Caching is Illustrated in Three Phases: Caching Placement, Replacement and Coherency Schemes. Caching placement Schemes are Further Arranged into Three Categories: Content-Based Caching (CBC), Content and Node-Based Caching (CNBC) and Alternative Caching Approaches

the following subsections, we present attempts that consider freshness in ICN-based caching design for IoTs.

a) Specific freshness Caching: In [80], freshness-based caching scheme is proposed to facilitate consumer applications inquiring contents with specific freshness values. Consumer has to specify the freshness requirement of the value it needs. Intermediate routers or producer can set (or even can change) the freshness value for the required content raising DoS attack. In CS, a new field to set freshness and a check to compare the time stamp of cached data with the requested by consumer have been added to the existing CCN. Consumer is assumed to send request for same content and with specific freshness values. Interest packet has been modified by adding a new field freshness parameter. Producer nodes are Wi-Fi nodes connected to Access Points (AP). LRU has been applied as cache replacement strategy. Freshness value added more control of the consumer in the quality of data being fetched. By adding, ratio of active time of restrictive in freshness consumer to active time of less restrictive in freshness consumer, caching performed better for IoT applications that need recent data. However in [80] only caching scheme has been presented.

b) Caching with same freshness: In [81], IoT environment needs and corresponding ICN features have been discussed. Bandwidth and energy consumption have been measured for CCN-based IoT scenarios with varying number of nodes (both consumers and producers) and compared against IP. CCN data packets have been modified by including both freshness of content and fraction of size of CS. NS3 and ndnSIM have been used for IP and CCN respectively. Application for consumer has been implemented in the way that it requests for same data from different producers. Total one hundred nodes have been included in the simulation while half of these nodes were producers and half were consumers. IP-based producers were WiFi mobile nodes connected to AP, while ICN consumer nodes were set to inquire data of same freshness value. LRU has been applied as cache replacement scheme and cache placement scheme has been designed to include freshness and variable CS size fraction. Impact of increasing sensors require more bandwidth rather than increasing number of consumers. This is good for IoT scenario where number of consumers are uncontrollable (e.g., hotspot or flash-crowd). They have found that IP-based case consumed more bandwidth than CCN. Impact of freshness has reduced performance assumed to achieve through caching. To enforce caching small CS would be enough if freshness is highly required. However, considered IoT scenario has fixed number of nodes and implementation has not been performed for dynamic IoT scenario.

2) Probability of content: Some IoT applications that require mix contents from multiple or single producer(s) like

Table VI
CACHING SCHEMES FOR ICN-BASED IOTS ACCORDING TO THE CLASSIFICATION PRESENTED IN FIG. 6. CBC IS FOR CONTENT-BASED CACHING AND CNBC IS FOR CONTENT AND NODE-BASED CACHING.

Reference	Placement Sub-Category Scheme	Replacement Scheme	Architecture	Comparison	Parameters Evaluated	Simulator
CBC Placement Schemes for ICN-IoT						
[80]	Different Freshness	LRU	CCN	IP	1.BW Consumption 2.Energy Consumption	ndnSIM for CCN and NS3 for IP
[81]	Same Freshness	LRU	CCN	-	1.Cache Hit Ratio 2.Avg. number of hops	ndnSIM and NS-3 for CCN
[82]	Dynamic Probability	LRU, Random	NDN	1.Always Caching 2.Probabilistic Caching	1.Hit Ratio 2.Retrieval Delay 3.Interest Re-transmission.	ndnSIM and NS-3 for NDN
[76]-[82]	Constant Probability (One Probability)	-	CCN	1.Always caching 2.No caching	Number of packets sent(Interest and Data)	RIOT OS
CNBC Schemes for ICN-IoT						
[84]	Freshness and Node Properties	LRU	NDN	1.No Caching 2.P(.5) Caching 3.Cache each and everything	1.Hit Ratio 2.Network Life Time 3.Retrieval Delay	ndnSIM and NS-3 for NDN
[85]	Popularity and Node Properties	Not mentioned	Not mentioned	Not mentioned	1.Cost Saving Ratio 2.Hop Distance Ratio	MatLAB for Analytical Modeling
Alternative Caching Schemes for ICN-IoT						
[86]	Infrastructure Based Caching	LRU	ICN	1.LCE 2.LCD 3.Prob Caching 4.Betweenness Centrality (Btw) 5.Client Cache With Zipf distribution	1.Percentage of validity 2.Response Latency 3.Hop Reduction Ratio 4.Server Hit Reduction Ratio	Analytical Modeling Simulator Not Mentioned
[83]	Infrastructure Based Caching	LFU in edge routers and LRU in centralized nodes	CCN COMBO project FP7	Current Transparent caching	1.No.of interests sent towards producer Vs towards cache 2.Play-back continuity 3.Average Latency	OMNET ++

in smart traffic, a car owner may be interested in the traffic condition ahead, temperature of that area, exact location of the vehicle and map towards its destination. Therefore, ICN-based caching strategies for IoTs should include factors to cope these applications requirements. In this context, random probability assignment can provide diversity in cached contents.

a) Always and Probabilistic Caching: In [82] authors have implemented NDN for IoTs and applied Always and Probabilistic (with P=0.5) caching schemes. LRU and Random replacement algorithms have been applied as cache replacement schemes. Simulations were performed in ndnSIM and NS-3. Total of 36 nodes were included in simulation, out of which, four were destined as consumers and six were randomly selected as producers in a 400m X 400m area. Probabilistic caching scheme and LRU cache replacement scheme, in a combination, achieved higher results for cache hit ratio, retrieval delay and interest re-transmissions. Cache size has been varied from 1-4KB but optimal results were achieved when CS size was 4KB. Probabilistic caching and LRU replacement scheme ensured content diversity and most recent contents in the IoT network, that are important requirements of IoTs. Though, authors have found caching (even with small CS) beneficial for IoTs.

In [76] impact of Always caching (Where P is always 1), is evaluated on RIOT OS [87] for a large building. They argue through their results that caching is highly beneficial for devices having small memory. Authors support in-network caching for IoTs because it saves bandwidth and energy consumption.

B. Content and Node-Based Caching (CNBC) for ICN-IoT

In this sub-section, we survey ICN-based caching schemes that include both content and node parameters. Content properties like freshness, popularity and node important parameters like battery level, cache size, node location and role in the network are considered for constraint-oriented IoT devices.

a) Probability of Freshness and Node Properties-Based Caching: In [84], authors presented probabilistic CAching STrategy for the INternet of thinGs (pCASTING), a caching mechanism considering content property (freshness) and node properties (battery level and cache occupancy). For caching replacement, LRU has been implemented. pCASTING has been compared against cache each and everything (CEE), probabilistic caching (P=0.5) and without caching. Simulations were performed in ndnSIM and NS-3. Total 60 mobile nodes were included in the scenario. There was only one producer and eight consumers were selected. pCASTING achieved higher cache hit ratio and received data packets by consumer. Retrieval delays were less than probabilistic and no caching but higher than CEE. However, only one producer has been assumed to reply. Popularity of content was not present in the cache decision.

b) Popularity and Node Properties-Based Caching: In [85] a caching scheme has been proposed using data freshness, request rate and router properties. Routers has been assigned the task to compute the probability of content, using content properties (freshness and request rate (popularity)) and node properties (incoming request rate and location of node in network). Numerical evaluation has been presented in Matlab. However, proposed caching scheme is for multimedia contents (40GB link has been mentioned in simulation parameters) and it requires extensive calculations, hence it is less suitable for IoT low power, constraint oriented devices to perform such complex and power-consuming calculations. Moreover, as mobile nodes change locations frequently (network topology changes), proposed method is highly suitable for static devices. As static devices do not face battery issues to perform such extensive calculations. However, they have not discussed about any caching replacement algorithm.

C. Alternative Caching Schemes for ICN-IoTs

In this section, we provide a comprehensive overview of caching schemes that do not focus on a particular method (i.e., content or node-based caching) but present caching schemes for IoTs from other perspectives. We categorize these ICN-based caching methods for IoTs into overlay caching and cache coherency schemes because they provide caching network architecture on the existing Internet and cache coherency mechanism for ICN-IoTs. Although ICN-based caching-node-architecture presented in [79], is not specifically for IoTs, but we include it to cope with the IoTs disaster management.

1) Overlay Caching for ICN-IoTs: An overlay shared caching scheme based on ICN is presented in [83]. A content management (CM) layer is introduced in Fixed and Mobile Converged (FMC) network architecture. This CM layer can be controlled through network provider or content producer. CM layer decides where content can be cached using its cache and metadata management schemes. Unified Access Gateway (UAG) node stores and forwards the content to any requesting node in FMC network while network is responsible for transmission of content. A cache controller (CC) is integrated in UAG that provides optimal caching and pre-fetching plans. HTTP traffic passes through this overlay caching. A *Config* packet is added in the CCNx to carry information about caching and cache replacement scheme. Updated CCNx provides transparent overlay caching and in pre-fetching process CC sends *Config* packet to cache node and which in return sends *Interest* message to overlay cache and overlay cache respond with *Data* packet. To provide mobility, they used BonnMotion [88]. Better performance of system is achieved in terms of, less number of packets sent towards original server as more packets get response from overlay caching, average latency and uninterrupted playback than the current system. Presented caching strategy and management scheme offers Caching as a Service (CaaS).

2) Client-Cache and Cache Coherency for ICN-IoTs: The work in [86] presents, an ICN-based cache coherence algorithm and a client-based caching strategy for M2M. Client-cache is named to represent the fact that content is saved in node near to the client node. Authors proposed client-based on-path caching strategy with less number of nodes and by using nodes that were close to receiver. A cache coherence algorithm has been presented to check the validity of contents. Proposed cache coherence method used expiration-based coherence with variable time expiration for every content. Client-based caching strategy was compared against Leave Copy Everywhere (LCE), Leave Copy Down (LCD), Probability caching, Betweenness Centrality. Client caching along with coherence algorithm has achieved better results in terms of hop reduction ratio, server hit reduction ratio, response latency and validity percentage of contents. To the best of our knowledge, this is only one paper that investigate cache coherency for ICN-based IoTs. However, cache size, that is selected, is much larger to suit for low memory devices to hold a large amount of contents. Moreover, discussion about IoT applications that require fresh content is missing in the proposed method.

3) Caching Node Architecture for Disaster Management: Authors in [79] consider the disaster situation and presented the solution to recover data through cache enabled nodes. A caching scheme is presented to collect fragmented data when network is fragmented or some device (producer) has left the current network. They have modified traditional CCN by introducing Satisfied Interest Table (SIT). An expression is presented to show until when content can be available and calculate its disappearance time. It is specifically designed when producer is moved and network got fragmented (disruptive Scenario). They tried to prolong a content availability through in network caching. Connectivity between friends and family is more crucial and bulk of data is produced in such situations. NDN router architecture is modified by augmentation of SIT. SIT will keep track of users with same interests and got required data. SIT will forward interest packet to users on the basis of entries it has saved. SIT entries are erased only when that user left the network. Interest packet is modified to be satisfied by producer or satisfied consumer by introducing Distention Flag (DF). If DF is 1 SIT will provide the satisfied user with same interest and now will provide the data against requested interest. Data Packet is same as of NDN. However, this scheme requires a lot of memory so it is natively not suitable for IoT small devices. But intrinsically suitable for nodes with excessive memory and it can be employed somewhere in IoT networks (e.g., as a backup nodes in IoT disaster management applications). It requires other users willingness to disseminate data and respond queries that can put a lot of burden on the network management and can raise security issues.

D. Summary and Insights

We have surveyed ICN-based caching schemes in the context of IoTs and provided a classification in Fig. 6. We have broadly categorized ICN-IOT caching mechanism into three phases: caching placement, replacement and coherency phases. Caching schemes have further categorized into three strategies: CBC, CNBC and alternative caching.

CBC schemes compute properties for every content, which include freshness and popularity of content. Researchers have

put more focus on exploring the content freshness while popularity has been explored in few approaches. Therefore, ICN-based content popularity caching for IoTs, seeks urgent attention from research community.

On the other hand, it is important to consider both node and content properties while making cache decision. On this side, a few efforts have been made to combine both features in cache placement strategies [84]-[85]. For this type of caching we categorize it into CNBC strategies. CNBC strategies include content properties along with IoT node characteristics like battery timings, CS size, node position and caching module designing in the node and IoT network type. As IoT nodes assumed to have low processing power, memory and battery. However, caching current literature is missing IoTs low power and low memory characteristics of nodes and IoT applications with moving devices. Moreover, caching strategies are lacking in push traffic type consideration for IoT network.

In comparison to decide about optimal caching schemes in ICN-based IoTs, CNBC is better than CBC alone in terms of throughput but obviously it requires more resources to compute about caching decision. ICN-based energy efficient caching schemes for IoTs are also needed to explore by research community.

Besides both CBC and CNBC, we categorize remaining ICN-based caching schemes for IoTs into alternative caching schemes. This include application specific caching node architecture like disaster management application, cache coherency protocol and overlay caching. This third category is decided independent of both node and content properties.

The survey proves that CBC has been explored to some more extent than CNBC. This is because CBC protocols directly deal with content properties like freshness and popularity. As every IoT application demands contents with different properties, for example, real-time applications demand highly fresh contents while flash crowds need more popular contents. As a result, CBC schemes are easy to explore for IoTs application scenarios. On the other hand, CNBC schemes are somewhat difficult to implement as ICN-based IoT node and network architecture are still under research and construction phase.

In caching replacement strategies, mostly LRU has been implemented in normal nodes due to its better results. While LFU has been considered for edge nodes. Random replacement scheme is easy and simple to implement that ensures high data diversity as well.

So far, there is only one cache coherency protocol for ICN-based IoTs [86], thus ICN-based coherency protocols for IoTs are urgently required to provide content validation in IoT applications.

In the nutshell, our extensive survey of ICN-IoT caching schemes indicates that ICN caching provides better IoT network performance and improves data delivery. Future research needs to explore CNBC caching schemes for IoTs constraint oriented nodes while accommodating both transient and ephemeral contents.

IV. 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 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 [21]-[20].

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 [89] connected devices efficiently. According to CISCO report, there will be 12.2 Billion IoTs smart and constraint-oriented connected devices in 2020 [90]. In addition, IoT network architecture is assumed to support scalability and heterogeneity.

Mainly there are two naming techniques (hierarchical naming structure and flat/hash naming) that are available through ICN architectures. CCN [91] / NDN [92] name contents in hierarchical manner while other ICN approaches (DONA [74], PURSUIT [93], COMET [14], MobilityFirst [16], SAIL [94] and CONVERGENCE [15]) follow flat self-certifying names. Third naming scheme, attribute-based has been used initially in CBCB (Combined Broadcast and Content-Based) routing [95] and can be used in combination with prior two naming techniques [96]-[97]. However, most of the research efforts considered and explored hierarchical naming technique for IoTs [98]-[99]-[100]-[76]-[101]-[22]-[34]. Some researchers focus on hybrid naming schemes incorporating both hierarchical and flat with attribute-based naming [102]-[103]. We categorized ICN-IoT naming schemes into four types which can be visualized in Fig. 7.

Therefore, naming IoT (devices and) contents through ICN ensure, efficient addressing and scalability, more security, better mobility and support for heterogeneous devices [29]-[34].

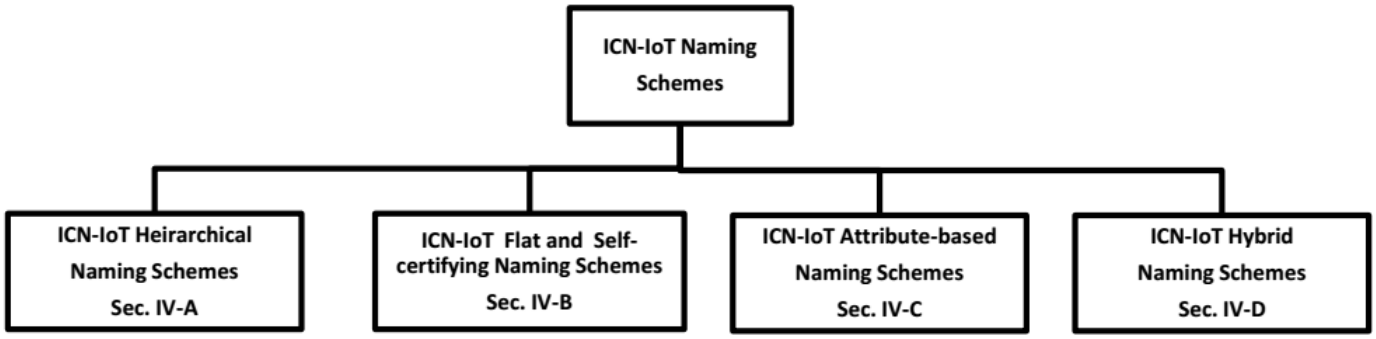

Figure 7. ICN-IoT Naming is Categorized into Four Categories: ICN-IoT Hierarchical Naming Schemes, ICN-IoT Flat and Self-Certifying Naming Schemes, ICN-IoT Attribute-based Naming Schemes and ICN-IoT Hybrid Naming Schemes

A. 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[96]-[97].

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 [100]-[104] 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 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 [105] 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 [22] 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 [98]. 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 [76] 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 [101] 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.

B. Flat Self-certifying-based ICN-IoT Naming

ICN native approaches like DONA [74], MobilityFirst [16] and *NetInf* [13] 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 [106], 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 [107] 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 [111], authors survey ICN architecture naming schemes and argue that self-certifying names provide name-persistence, security-binding and universal uniqueness. Moreover, in [112] 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[113]-[114].

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.

C. Attribute-based ICN-IoT Naming

This naming approach extract attributes of content and was used initially in CBCB [95]. 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 [115] using attributes of the content. In [108], 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 [111], 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.

D. 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 [96]-[97].

In [102] 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.

Table VII
ICN-BASED IOT NAMING SCHEMES ARE SUMMARIZED ACCORDING TO THE FIG. 7. HERE NLAPB IS FOR NAME LOOKUP SOLUTION WITH ADAPTIVE PREFIX BLOOM FILTER.

Reference	Architecture	Comparison	Parameters Evaluated	IoT Application	Simulator (OS, Programming Platform, Language)
Hierarchical Naming Schemes for ICN-IoT					
[104]-[100]	CCNx	IP	1.Retrieval Delay with and without caching 2.Number of Exchanged Messages	Temperature Measurement Wireless Sensor Networks	Contiki OS and Cooja Simulator
[76]	CCNx	6LoWPAN/RPL/UDP 1.Vanilla Interest Flooding (VIF) VS. Reactive Optimistic Name -based Routing (RONR)	Number of Consumers VS. Number of Messages Sent (With and without Caching)	Building Automation	RIOT OS
[22]	NDN	-	1.Number of transmission(s) 2.Number of Exchanged essages Vs Number of producers	Smart Home	No simulations Not mentioned
[98]	NDN	-	No simulations Not mentioned	Light Control System (Instrumented Environment)	Not mentioned
[101]	NDN	-	No simulations Not mentioned	Building Management Systems	Python-based Application Java-Scripting Data Visualization Application
Flat (and Self-Certifying) Naming Schemes for ICN-IoT					
[106]	CCNx	IP-based WSN	1.Average energy consumption 2.Average delay	WSN	Contiki OS and Cooja Simulator
[107]	ICN	Not provided	Not provided	Not for low-power IoT devices	No Simulations Not mentioned
Attribute-based Naming Schemes for ICN-IoT					
[108]	ICN	With and without ontology	1. Storage Overhead 2. Transfer Time Consumption	Smart Hospital	C Language
Hybrid Naming Schemes for ICN-IoT					
[102]	NDN	No Comparison	-	Vehicular Ad-hoc Networks	No Simulations Not mentioned
[109]	NDN	No naming Comparison	1.Start-up delay 2.Playback Freezing Ratio	Multimedia Contents dissemination in VANETs	NS3 with ndnSim
[103]	NDN	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	Vehicular Ad-hoc Networks	Not mentioned
[110]	CCN	Hierarchical and flat naming aggregation	1.Interest transmission rate 2. Number of covered hops and exchanged messages	IoT Smart Campus	Contiki OS with Cooja Sim

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 [109], 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 [103], authors investigated hybrid naming scheme proposed in [102] 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 [116] 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 72bytes. 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 (i.e., Payload size is 127 Bytes).

In [110], 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.

E. Summary and Insights

In this section, 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 survey 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 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.

V. ICN-IoT Security Schemes

In today's Internet and IoT applications, security is a basic need and a central factor from design perspective. Because almost all IoT applications tend to take data from our daily life gadgets and involve third parties to process that data creating a potential to affect our privacy. As content security was not inherited in IP-based Internet applications but security features like content integrity and device authentication are added later as an add-on. IP-based protocols like EAP, PANA, SSL, DTLS and IPv6-based security solutions employ location of nodes. These security protocols secures communication channel between nodes instead of content. By adding security as a patch on IP, constraint-oriented IoT nodes perform with delays. Handling of mobile devices complicates the situation even more. Moreover, IoT system is completely secured when it ensure authentication, authorization, confidentiality and integrity.

While ICN offers security at network layer and provides communication on the basis of contents. Content-based security provides easy and simple security to IoT contents without involvement of third-parties or external intermediate nodes. Content-based security maintains content integrity and data

authentication. Moreover, ICN contents can specify content access control towards users due to the fact that ICN contents are generally known as self-certified contents.

We categorize ICN-IoT (ICN-based IoT) security schemes into following three categories: (i). ICN-IoT device security schemes, (ii). ICN-IoT content security schemes and (iii). ICN-IoT content and device security schemes. ICN-IoT device security schemes deals with device authorization and authentication. While ICN-IoT content security schemes provide content integrity and confidentiality. N ext, b oth c ontent and devices are secured by ICN-IoT content and device security schemes. Categorization in ICN-based security for IoT is visualized in Fig. 8.

A. ICN-IoT Device Security Schemes

In [117], ICN-based secure protocol is proposed which provides security in terms of both authentication and authorization for IoT devices. They call this ICN-based security protocol as on-boarding protocol (OnboardICNg). OnboardICNg protocol authenticates every joining device and authorize it through authorizing this device. They consider authentication and authorization manager (AAM) for initial key sharing. Key is shared between new joining device and AAM to guarantee it as a secure IoT device. The new device knows the naming format of publishing and requesting any content. A single key is supposed/assumed to provide authentication, integrity and confidentiality. They used and modified, authenticated key exchanged protocol (AKEP2) according to the ICN design for IoTs. Through OnboardICNg, IoT network is secured from internal and outsider adversaries. They compare OnboardICNg with Pre-Shared Key Extensible Authentication Protocol (EAP-PSK)/PANA in terms of communication cost (both communication and computation costs) and energy cost (both energy and memory costs). They find OnboardICNg is more effective for IoTs with 87% and 66% reduction in communication and energy costs respectively as compared to EAP-PSK/PANA. However, authors do not provide any simulations and present only analytical results for the proposed protocol.

Authors in [118] enhances Onboarding authentication protocol and combines routing with it. They call proposed protocol lightweight authentication and secure routing (LASeR) protocol. They consider islands making IoT smart cities. Considered scenario have anchor nodes, standard nodes and gateway nodes. Among which standard nodes are IoT nodes only. An island manager (IM) just like AAM in [117] is used to authenticate and authorize the nodes. LASeR protocol works in three steps: discovery phase, authentication phase and advertisement phase. They evaluated LASeR in terms of convergence time and transmission burden for different number of nodes and increasing distances among nodes. LASeR only focuses on authentication with routing. However, IoT nodes does not involve in this whole procedure, they delegate their duties to anchor nodes and IM. Like [117] they also talk about securing the IoT applications and nodes as a whole.

B. ICN-IoT Content Security Schemes

In [103] authors have presented secured content naming scheme where content name is secured using Base64 Format. This work is performed for multimedia contents fetched by vehicles. The secured part is included at the end of Interest packet and can be calculated by taking hash of attributes of content or public key of the vehicle. They have programmed it in Linux-based C++ programming. They have only consider vehicles and not static devices.

In [119] we propose a IoT content naming scheme. IoT applications categorization is updated and a universal hybrid naming scheme is proposed. Content is secured using SHA256 to maintain integrity. Fetched content name and its sub-type name is encrypted through SHA256. Moreover, name of the node that is originating the Interest is also encrypted through SHA256. Security is preserved in the context of integrity. However, no implementation is presented.

C. ICN-IoT Content and Device Security Schemes

To secure buildings, NDN-based architecture is presented in [101] and it is installed in University of California at Los Angeles (UCLA). This is just a prototype to show the performance achieved by NDN instead of IP-based security systems. Their proposal consists of three main entities, end users, gateway and a manager application. Gateway and sensor devices run IP-based building management system (BMS) protocols. Manager application is controlled by a human operator and authorizes out of band users. It is also responsible for NDN management and auto-configuration of sensors and gateway. Gateways publish contents into NDN repositories. NDN repositories are responsible to respond user queries about sensors data. In NDN-based BMS, they follow and designed hierarchical naming to name devices and contents. They used public keys of any user and append it as last component of content name by calculating its hash through SHA256. To maintain user privileges two list are maintained. Each gateway has access control list (ACL), which is a list of identities of authorized users. Another list, access privilege list (APL) contains the data names-paces that any user can access and is maintained by every user of BMS. APL is also published in NDN repositories. To provide mapping between content namespaces and user IDs, both lists (i.e., ACL and APL) are responsible. This saves BMS manager from traversing entire BMS application to update user privileges. Both ACL and APL can be published as NDN data. They consider capability-based access control. ACL lists the capabilities to access sensor data and user gets capability-certificate to access data. During gateway configuration, NDN packets are signed and encrypted using symmetric key to secure from man-in-middle attacks. Sensor data is encrypted through shared symmetric key to provide access-control and published in JSON format. Gateway generates and distributes symmetric keys while going through ACL. It publishes encrypted key (encrypted through user's public key) asymmetrically. Data packet also contains time-stamp of decryption key to ensure content-based security. Python-based data publishing service is used to publish data and browser-based data visualization

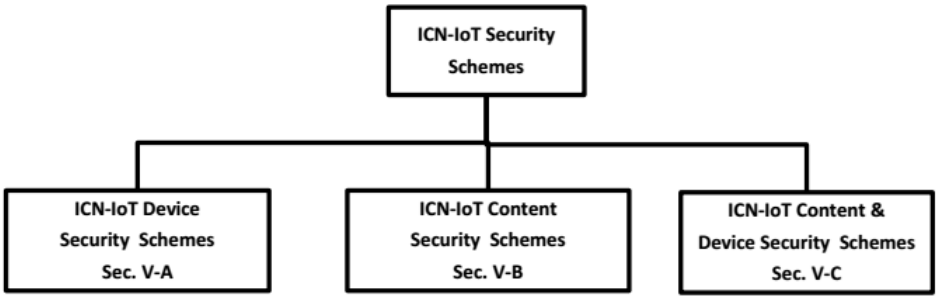

Figure 8. ICN-IoT Security is Categorized into Three Categories: ICN-IoT Device Security Schemes, ICN-IoT Content Security Schemes and ICN-IoT Content and Device Security Schemes

application. The data publishing service packs data in JSON format into NDN repositories. User issues interests using data visualization application and can employ time-stamp filter. It gets encrypted data and decryption is performed through encrypted symmetric key. Data is encrypted using AES-CBC cipher. The BMS system presented in this asynchronous approach is not suitable for IoT a situation where fresh data is required from a sensor because sensor uploads data into NDN repositories first. However, it enables caching, lowers load on data server and preserves IoT scalability as data is secured via encryption only single time.

In [120] authors discuss forwarding and security for ICN-based IoT. Geographic forwarding is implemented due to its low control traffic for sending data towards destination. It involves location of destination for content transmission and thus lower network resources usage while maximizing energy life of IoT devices. To provide security, authors force the use of symmetric cryptography through OnboardICNg. They state that OnboardICNg authenticates locally two nodes and verifies that both are parts of a trusted network. Through provided shared symmetric key, nodes authenticate each other to build a secured network. Next they discuss, secure push mode through secure beaconing. Insecure beaconing can introduce DoS and wormhole attacks. Through broadcasted shared symmetric keys, sensors distinguish the beacons from the trusted users. Beacon messages are secured by encrypting these through the broadcast keys provided by OnboardICNg. Further messages after beacon, contain MACs generated through encryption using broadcast keys. However, if neighboring node is tempered then the scheme is not resilient. They evaluate their proposal in RIOT OS in terms of computation, network and memory footprints. It takes 28 to 35 extra bytes per message like beacon, interest and data message during transmission in 802.15.4-based OpenMote. AES-CCM takes more energy both in software and hardware, it is one order lower than transmission of messages. Cost of memory footprints includes three keys per node and authors state that this is likely a negligible space available on most recent boards like OpenMote. However the main aim of this proposal is to evaluate geographic forwarding in ICN-based IoT. They also evaluate OnboardICNg on both hardware and software and find that security comes at a cost. This proposal secures ICN-based IoTs through securing IoT devices and contents.

In [121], authors discuss benefits and challenges of applying ICN for IoT. They consider two content requests, (i) when any user wants an action performed by any device and (ii) when user requests the current content of the device. Their proposal consists of gateway, admin, clients with same name-space, IoT devices and other clients. Gateway is the central device which connects with admin, IoT devices and clients to provide interoperability between powerful and constraint-oriented devices. This gateway is also placed to cope with heterogeneous devices differentiated as devices from different name-spaces. Gateway exchange, management content information, with IoT devices through the reference point Mdg. This Mdg as reference point is responsible for secure content centric communication with IoT devices. Client and gateway mutually authenticate the security mechanism for full proof content exchange in CCN. Through discovery procedure, client discover a list of IoT devices. In its working, as step 1, client first expresses an interest in the form of CCN name. In step 2, gateway receive this interest and respond with data packet. Data packet indicates content protection and also provide information to client for encryption algorithm and key sizes. For normal CCN phenomenon, data also incorporate shorthand identifier for the gateway (i.e. GW publisher ID). GW publisher ID is calculated through cryptographic digest of its public key and key locator is responsible for actual location of public key. In step 3, in order to get appropriate key client issues an interest for the protection of exchange information. Then, client get verified through gateway to enable IoT service routine. When client is authenticated, gateway generates a random systematic key SKcg (128 bit AES key) for cryptographic functions. This SKcg key along with its related information are encrypted with public key of client as extracted from data packet in step 4. Data provided by gateway is verified and decrypted by client through its SKcg. Client also generates Message Authentication Code (MAC) over the whole interest by using the session key SKcg. In step 5, MAC and a unique nonce value is appended with CCN name to prevent malicious attacks. Gateway verifies nonce and MAC component and replies to interest message with data packet in step 6. As step 7, client can retrieve information from gateway by issuing interest after validation of client. Then gateway reply client in accordance with client specific policy in step 8. Gateway-based proposed design, presented in this paper have

Table VIII
ICN-IoT Security Schemes Are Summarized According to the Fig. 8.

Ref.	Model	Security Perspective	Methodology	Comparison	Parameters Evaluated	Finding(s)	Simulator (OS, Programming Platform, Language)
ICN-IoT Device Security Schemes							
[117]	ICN	Device Authentication Device Authorization	Symmetric Cryptography	ZigBee-IP specification: EAP-PSK/ PANA	Communication cost (communication and computation) and energy consumption (energy cost, memory cost)	87% less communication 66% energy consumption helps in confidentiality of content, which in turn maintain privacy	ANALYTICAL EVALUATION
[118]	NDN	Authentication Authorization Routing	Symmetric Cryptography and routing	With its own variants in terms of increasing number of nodes and distance	Probability Mass function, Transmission burden, convergence time	Light weight Authentication and secure routing	ndnSIM an ns-3 extension
ICN-IoT Content Security Schemes							
[103]	CCN	Integrity	Base64 Format on Content name	No Comparison	No Implementation	Maintains Integrity of content name and device name	Linux-based C++ programming language
[119]	ICN	Integrity	SHA256 on Content name	No Comparison	No Implementation	Maintains Integrity of content name and device name	No Implementation
ICN-IoT Content and Device Security Schemes							
[101]	NDN	Data Privacy Data Authentication	Data Privacy through Access Control Data Authentication through Digital Signature	No Comparison	Analytically Evaluated Data Scalability preserved	More responsive More scalable Less load as compared to IP-BMS	Python-based Application Data Visualization Application
[120]	ICN	Security and geographic forwarding	Secure Beaconing through OnboardICNg	Vanilla ICN forwarding	No. of FIB entries, energy cost, Network overhead, memory and computation overhead	OnboardICNg takes extra computation, energy and memory	RIOT OS
[121]	CCN	Device authentication Content Integrity	PK Cryptographic Suite Symmetric Encryption using AES	Arduino board for proof of concept	1.Info. Freshness level, 2.Interest Range stability 3. Energy consumption with or without security feature via UDP and CCN 4. Packet overhead estimation	1. Avg. Service time is stable for interest rate less than 24 request/s 2. Energy Consumption with security feature 0.33% Without security with CCN feature 0.28%	ndnSIM 1.0
[122]	ICN	Privacy, trust, content integrity, confidentiality, authentication, access control	device discovery service discovery secure subscription, Secure naming service, Secure content delivery	No Implementation	No Implementations	Secure ICN-IoT Architecture	UML diagrams

the flexibility to adopt according to according environment and organization. It also enable security feature through built-in support of automatic discovery and registration process that is the uniqueness of this design. It also reduce the overall incoming interest packets. Result shows that the average service time of interests is stable for 25 requests per second. This work provides is suitable for IoT as it can scale up with less overhead and secures both IoT contents and devices.

In [122] authors proposed an ICN-based secure architecture for IoT. Proposed ICN-IoT secure architecture provides trust model for nodes and links, privacy for sensitive information and effective access control system. Five components including IoT nodes (Content producers), service consumers, ICN-IoT server, local server gateway (LSG) and aggregator, build proposed ICN-IoT middleware. They integrate security with ICN-IoT architecture [123] interactions involving, device discovery, service discovery, naming service, user registration and content delivery. Authentication of devices is performed through device discovery phase. Secure device discovery is ensured when any new device joining IoT network send its device ID, signature key and certificate; this triplet is sent towards aggregator where it verifies and stores this new device information. Then aggregator issues a action key encrypted through signature key. If the new joining device is not a certified device then it can send its device ID only. In this case, aggregator can issue signature key and certificate. This method can be helpful for mobile devices authentication. Further service discovery is used by IoT users to get any service.

IoT user connects with ICN-IoT server through sharing its both signature key and device ID. Upon successful access grant, user further send its actual query/request in encrypted form through its action key and signature key. ICN-IoT server forwards this request towards aggregator. Aggregator decrypts and satisfies request with the help of IoT nodes and sends relevant response towards corresponding IoT user. Secure naming service provides security to names of IoT devices. Aggregator sends device ID, signature key and action keys towards LSG which in turn assigns the name to device and replies name to aggregator. Aggregator sends device name towards device by encrypting it through action key of the device. During a subscription, user needs a secure subscription and it is performed through secure user registration. User contacts ICN-IoT server by sharing its own information along with device name. ICN-IoT server replies user with ID, signature key and password (which user can change). Secure content delivery from device is ensured by sending device name, ID encrypted with signature key and data encrypted with action key to aggregator. Aggregator decrypts data and sends to ICN-IoT server. ICN-IoT server again encrypts data with action key of the user and sends towards user. Proposed ICN-IoT architecture aims to secure both content and device by maintaining privacy, authentication, confidentiality and integrity. However, authors didn't provide simulations to verify the results. They only provide UML diagrams to describe their proposal.

D. Summary and Insights

In this section, we have surveyed ICN-based security schemes in terms of IoT and classified these security approaches into three categories. In first category, we listed and summarize those approaches which handle ICN-based security of IoT devices. These approaches mainly provide authentication and authorization of IoT devices. Second category, ICN-IoT content-based security schemes mainly deal with content and aimed to provide content integrity, non-repudiation and confidentiality. The resulting contents are self-certified which can specify its owner details and content details. In third category, ICN-IoT content and device security schemes, those approaches are discussed which include both device and content properties. ICN security approaches in this class mainly focus to secure the whole IoT system while providing content integrity, confidentiality and device authentication and authorization. Moreover, some techniques also added access-control-management which aimed to specify the list of intended users.

Our survey finds that ICN-based security schemes must be designed that involve IoT environment characteristics; for example, considering constraint-oriented nature of IoT devices. As IoT applications can involve push operations; for instance, an actuator IoT device can only perform a simple action like turning some devices on/off if this query/command is received from authenticated and trusted IoT node. But most methods discussed above apply security methods over interest and data messages. Therefore, there is need to ensure that security mechanisms must provide authenticated requests along with enabled push support.

Moreover, public key cryptography (asymmetric cryptography) can not be implemented for IoT resource-constraint (i.e., in terms of memory and processing) devices because of its resource-intensive nature. ICN-IoT content security schemes which embeds security information at the end of query/interest packets as last named component, result lengthy request packets and increase complexity to be processed by IoT constraint-oriented nodes. For this reason, lightweight security solutions to maintain confidentiality, integrity and authentication are optimal and feasible choices for IoT constraint-oriented nature.

From this perspective, symmetric key cryptography can play important part and is explored in many approaches like [101]-[99]-[117]-[118]. As, symmetric cryptography approaches need to maintain keys and exchange of these keys is required before any communication. However, these pre-shared keys cause extra overhead and makes symmetric key cryptography inflexible for IoT.

Besides these, now-a-days Elliptic Curve Cryptography (ECC) is being explored for IoT constraint-oriented devices because of its simplicity and extra lightweight nature. ECC utilises elliptic curve theory to produce better cryptographic keys in terms of size and efficiency. As compared to RSA algorithm, where the keys are generated from the product of two large prime numbers, ECC creates them through the properties of elliptic curve equation. It relies on the difficulty of solving the elliptic curve discrete logarithmic problem. Although the key size in ECC is smaller, it can provide as good security as any other traditional method such as RSA which eventually reduces the processing cost. Therefore, it is expected from ECC to provide essential security features for secured ICN-based IoT.

Finally, to conclude, our survey of ICN-IoT security schemes indicates that there is no single solution that fulfills all requirements of IoT nodes and applications. Therefore, ICN-based IoT security solutions must be designed in a flexible way that include both IoT application requirements and IoT devices specifications and capabilities.

VI. ICN-IoT Mobility Schemes

As IoT networks can include hybrid and heterogeneous devices in terms of mobile and non-mobile (i.e., static) devices. While most of the IoT applications such as smart home, smart grid, smart building require mostly static devices. But other applications like smart transport, smart vehicles, smart mobile networks involve more mobile devices as compared to static devices. Therefore, mobile devices are important part of IoT and thus their management also become essential.

Although there are other mobility models (like nomadic and pervasive) but in IoTs, cellular mobility model plays an important role. As in cellular mobility, wireless networks are divided in cells and each cell has specific radius and area of service. While moving from one cell to the next, mobile devices face a situation called handoff condition. Thus handoff-management is also becomes an important factor to solve.

In ICN-based IoTs, both subscriber and producer can be mobile devices. As described and discussed before, ICN-IoT mobile subscriber can benefit from connection-less and

receiver-driven nature of ICN. In this way, mobile subscriber can re-issue interests for which they didn't receive data.

To support mobility, DTN function don't need heavy protocols like Mobile-IP. In contrast, publisher mobility is complex to manage as it requires some additional operations.

We categorized ICN-based mobility schemes into ICN-based IoT producer mobility management schemes and other ICN-IoT Mobility schemes. In first category, those schemes are combined in which ICN-based producer mobility is discussed. ICN producer mobility scheme further categorized into anchor-less producer mobility. In other producer mobility schemes, ICN-IoT smart forwarding schemes are discussed.

A. ICN-IoT Producer Mobility

Producer mobility is accomplished in two steps. Firstly it is needed to find and track producer location along with graceful session maintenance. Producer mobility handling generally depends upon that the architecture is coupled or decoupled in terms of name-resolution and data-transfer. In coupled architecture, producer advertises content prefix from its new location. While in decoupled approaches, resolution information is needed to update from new location.

In [124] producer mobility support mechanisms and their disadvantages are discussed in three categories. Routing-based producer mobility is provided by updating the routing tables that involve the forwarding of information queries. However, routing-based approach is not suitable to provide scalability of routing tables. Second, indirection approach requires some extra nodes (home-agents) which keep track of nodes locations and forward interests to the updated location of mobile producer. Drop-acts of this approach lies in the form of extra management of content names and their name-resolution (i.e., information of producers), and every query and data message also visit this home-agent. Third approach, resolution-based include content updated location (or information about updated location) in data message as response of user query. Resolution based approach incurs overhead of this one extra packet. This work discuss the feasibility of ICN mobility in terms of both mobile producers and consumers in opportunistic and mobile networks which is a definite part of ICN-IoT. They further discuss both content discovery and transfer mechanisms.

In [125] NDN-based producer mobility is discussed for IoT. They discussed NDN-based producer mobility support through four approaches. First approach solves producer mobility by utilizing the location information through location resolution system (LRS). Producer updates LRS about its location after moving. LRS keeps record of content name prefix and its corresponding producer. Consumer requests the location of content producer by sending the message having content prefix towards LRS. In second triangular approach, interest message is sent towards previous location and using FIB update, it is rerouted towards new location. Data message is delivered firstly towards old location and then from there, it is forwarded to consumer. In third locator/identifier separation approach, every content is managed in two parts by its producer. Content first part is its identifier and second is its locator. In identifier, prefix or content name is stored and in locator, location of the router (to which it is currently connected) is saved. After producer mobility, it changes its locator value with the location of new connected router. Fourth approach, routing-based approach finds the data through name-base routing protocol. Name-base routing protocol tries to find the cached copies of data towards the path of original producer. Name-base routing can be implemented through decentralized routing using flooding and distance-based greedy routing protocol. And thus its complexity depends on routing protocol. They expect that name-based routing scheme can perform better in IoT due to its medium cost for packet delivery, less handover latency and optimal routing patch length. However, they didn't propose any technique for NDN-IoT producer mobility.

In [126], authors surveys producer mobility and categorize into four categories: (i) mobile producer (MP) mapping, (ii) MP tracing (iii) data depot and (iv) data spot. In MP mapping, MP informs rendezvous (RV) node about its point of attachment (PoA) and data can be obtained through mapping provided by RV or RV tunnels the interest messages towards MP. In MP tracing, interest messages can use traces (if meet any) of MP on the way towards RV and get forwarded towards MP without involving RV. In data depot, a stationary location saves the data produced by MPs and can forward the data in response to interests with involving MP in this whole procedure. Finally, in data spot, new MPs generate data in order to fulfill the interest. However in IoT, data depot along with MP tracing (or mapping) plays the part due to nature of IoT applications. Moreover, data depot along with tracing can enhance interest satisfaction rate as IoT devices may run out of battery more oftenly and traces can provide direct path towards MP.

1) Anchor-less Producer Mobility: In [127], proposed producer mobility management (MM) scheme is designed to meet 5G requirements of low latency, low network overhead and overall fast speed. MM schemes are categorize into three classes: (i) anchor-based, (ii) anchor-less and (iii) rendezvous-based. In anchor-less MM, any node is responsible for providing information about its new location. In rendezvous-based MM, dedicated nodes are responsible for providing resolution of identifiers into locators. In third approach anchor-based, a specified node is responsible for all nodes movements and direct messages to the new locations of moved nodes. They have proposed anchor-less MM system to support delay-sensitive applications like smart health. When a patient is moving and acts as mobile producer, its fast MM is important. They used state-ful forwarding, ICN in-network caching and defined forwarding mechanism to update and populate Temporary FIB (TFIB) from producer new location towards its former location. MM does not need global routing updates and any change in the content name. It employs the distributed and dynamic ICN forwarding and eliminate the need for in-network anchors while limiting the MM towards edge nodes. Anchor-less MM is lightweight in nature because it limits signaling and maintains temporary change or state by in-network nodes. To support latency-sensitive transmissions during high mobility, network notifications and discovery methods provides necessary support. Anchor-less producer mobility is ensured in three simple following steps. Every

Table IX
ICN-IoT MOBILITY SCHEMES

Ref.	Model	Mobility Perspective	Methodology	Comparison	Parameters Evaluated	Finding(s)	Simulator (OS, Programming Platform, Language)
ICN-IoT Producer Mobility							
[124]	NDN	Producer Mobility	Content transfer content discovery	No Comparison	No Implementation	Delegating content retrieval to agents is better	CCNx
[125]	NDN	Producer Mobility	Survey	No Comparison	delivery cost path length interest routing	Name-based routing is better	Analytical Evaluation
[126]	NDN	Producer Mobility	Survey	No Comparison	Signal overhead security name-changes dependency on RV	data depot+tracing, data depot+mapping are better	Analytical Evaluation
Anchor-less ICN-IoT Producer Mobility							
[127]-[128]	NDN	Producer Mobility	IU and IN through Sequence Numbers	GR, AB, TB	Avg. Packet loss, delay & hop-count No. of messages, signaling overhead link utilization	Better network cost & user performance	ndnSIM
[129]	NDN	Secure Producer Mobility	Hash and Hash chains	MD-1,-5, SHA256, DSA, RSA	Computation Overhead Storage overhead	Lightweight attestation & Scalable	ndnSIM

mobile producer updates content (it produces) as a list of prefixes to its new PoA after establishing link with this PoA in a defined message called Interest Update (IU). After a relocation, producer changes router and populates TFIB using forwarding update operation. Consumer interest is forwarded towards producer using this TFIB information or using FIB along with discovery mechanism. In [128], they have evaluated their proposed anchor-less producer MM and called it Map-Me. For delay sensitive applications, producer left its traces on the way to its new location and they named it Interest Notification (IN). Due its lightweight nature, IN supports delay sensitive applications. They also provide both analytical and simulation evaluation. Simulation is carried in ndnSIM with total 36 wifi nodes. They found proposed MM better than global routing, tracing-based and anchor-based approaches in terms of average packet loss, average packet delay, average hop counts, number of messages, signaling overhead and link utilization. This anchor-less MM is highly suitable for IoT applications and delay sensitive applications like smart health.

In [129], authors identified loop-holes of [128] and propose a prefix attestation protocol to secure trace-based producer mobility. Protocol Map-Me can be compromised when IU came from any attacker. It can pollute cache and disturb privacy of consumers and edge routers. Session-key and signature based used for securing routers. However, both are not suitable for 5G networks. In their prefix attestation protocol, producer sends minimal security context towards registration server to generate valid IU. This security context is distribute locally among local routers and they use this information to validate IU locally. Security is maintained while allowing fast validation and generation of valid IUs through hash functions and hash chains, respectively. They evaluate attestation protocol analytically in terms of goodput. Goodput decreases when because IUs take resources. Hash chains maintains optimal goodput in case of one hash or multiple hashes per IU verification. Around 50 MB are required for millions of mobile users in one router and proposed prefix attestation protocol is thus more scalable.

B. Other approaches in ICN-IoT Mobility

In [130] a forwarding mechanism is presented for vehicles by incorporating one immediate vehicle resources. It ranks the vehicle based upon multiple factors and selects one as forwarder among all vehicles. However it doesn't account the provider mobility (i.e. adhesive issue of ICN mobility). Moreover, in [131], authors provide a scheme DPEL (Dynamic PIT Entry Lifetime) to reduce number of PIT entries. Hence it minimizes the usage of battery of mobile nodes and makes routing and forwarding easy and fast.

C. Summary and Insights

This section presents ICN-based IoT mobility and categorized presented schemes. As ICN supports consumer mobility naturally but mobile producer support is undefined. ICN consumer can re-issue interest for any missed packet and can get data after location change. ICN producer mobility is hard to handle.

As IoT needs fast data continuity in real-time applications. Moreover, resource-constrained nature of IoT devices put more challenges like tracking mobile devices in terms of old and new locations of mobile devices, reducing handover delay and simplify mobility management and handling with less number of packets. In this context, anchor-less producer MM [127]-[128] can be employed for IoT environment and can be secured further though hash chains method presented in [129].

Moreover, in other ICN-IoT schemes those schemes are included which try to make IoT mobile node lighter while

minimizing PIT entries and selects best forwarder among available vehicles.

However, there is not any single solution exists for ICN-IoT producer mobility and handoff management. This may be due to the fact that IoT general applications like smart home involve mostly static devices. Therefore, mobility is the most ignored perspective and available as fertile research direction.

VII. ICN-IoT Operating Systems and Simulation Tools

There are a lot of IoT Operating Systems (OS) and simulation tools that can be used for ICN-IoT. In [26] famous IoT OSs (Contiki [132], FreeRTOS [133], RIOT [87], TinyOS [134], OpenWSN [135]) are presented under the category of open-source and closed-source (that are not available commercially). Among them, we discuss only which can be used for both IoT as well as ICN implementations. On the other hand, specific I CN s imulators (ndnSim [136], ccnSim[137] and Icarus [138]) are presented in [139]. However, from this paper perspective, it can be seen in Table X that ndnSIM for NDN is the most explored simulator for ICN-IoT.

A. Contiki OS with Cooja Simulator

Contiki [26], [132] is an open source and flexible operating system developed at the Swedish Institute of Computer Science (SICS) in Sweden. It is very lightweight operating system for sensor nodes which are severely resource constrained in terms of power, memory, processing power and communication bandwidth. Contiki is developed in C language and is event driven. The main features of Contiki operating system include: the support of preemptive multithreading per-process and dynamic loading and unloading of code at run time. A Contiki configuration consumes 40 kilobytes of ROM and 2 kilobytes of RAM. The communication between different processes always goes using the kernel of operating system only. A full installation of Contiki operating system includes many features such as: preemptive multithreading, TCP/IP networking, proto-threads, Graphical User Interface, multitasking kernel, IPv6, web browser, simple telnet client, personal web server, and virtual network computing. Its current version is 3.0 released on August 26, 2015. Cooja Simulator [140] is the Contiki network simulator. Cooja allows large and small networks of Contiki motes to be simulated. Motes can be emulated at the hardware level, which is slower but allows precise inspection of the system behavior, or at a less detailed level, which is faster and allows simulation of larger networks. Contiki along with Cooja Simulator makes it a perfect combination for ICN-IoT related research.

B. RIOT OS

RIOT [87] is licensed as LGPL (Lesser General Public License) and open-source operating system for sensor nodes in the Internet of Things. RIOT OS is a microkernel-based operating system inherited from Fire Kernel [141], that matching the various software requirements for IoT devices. The key design objectives for RIOT OS include: energy-efficiency, small memory footprint, modularity, and a developer friendly programming interface, which make RIOT the best choice to power the widest spectrum of IoT devices. Implementation and design of RIOT has the ability to deals with the various challenges in powering of constrained devices networks. RIOT also provides the both real-time capabilities and full multi-threading. RIOT provides the C and C++ programming language supports for applications

C. Other Simulators

NDN architecture can be simulated using its own specific ndnSimSimulator. This ndnSim [136] is NS3-based simulator and provide simulation for NDN and ICN.

Mini-CCNx [142] is a tool for agile prototyping of ICN-based on the CCN model. This is use to build several CCN topologies, each with hundreds of nodes, with great agility and flexibility. These topologies can be run directly on laptop/desktop, in a local VM or in cloud. And the best is: the code you run on Mini-CCNx is the same code that you'll use in a real network. This really adds a realistic behavior to your tests. Each Mini-CCNx node (host or router) runs the official Project CCNx's so you'll be using the official CCN implementation.

ICN Simulator the Information-Centric Network Simulator developed by the University of Essex works with OMNET++ simulation environment. It provides PURSUIT architecture functionalities. It is able to simulate a large number of nodes and publisher-subscriber pairs and produce a huge amount of information, providing an insight on the new techniques introduced in the topology management of the information-centric network.

Icarus [138] is a caching simulator that supports multiple caching schemes and replacement schemes. It is Python-based and is a general tool to evaluate and implement ICN caching schemes. It does not support any specific ICN flavor but a simple environment to work with ICN caching.

VIII. Issues, Challenges, and Future Research Directions for ICN-IoTs

In this section, we present issues with the current solutions for ICN-IoTs and identify future research directions that need to be solved by the research community.

A. Naming

Most of the ICN-based IoT naming research is conducted for CCN/NDN hierarchical naming. As CCN header is of fixed size (8 bytes) [143]. Therefore, to apply CCNx (with fixed header) for IoT low-power and constraint-oriented devices, header compression techniques can be explored to support small data packets.

However, NDN packet [11]-[144] does not have fixed length header. For small data packets (like mostly IoT applications have short length data to transmit in response of a query or to send command towards any sensor or to just acknowledge the command to or to send current state of any sensor), NDN packet formats with variable length headers provide good support for IoTs applications [144].

Table X
ICN-IoT OS and Simulation Tools Analysis

	ndnSIM [136]	Contiki OS/Cooja Simulator [26], [132]	RIOT OS [87]
Ref. #	[80]-[81]-[82]-[84]-[109]-[118]-[128]	[100]-[104]-[110]	[76]-[120]
Total # of Ref.	7	3	2

In addition, as CCN/NDN naming follows hierarchical structure that generates long and variable length names, and these long names can be utilized to build applications that have to update their status (or sensor values) continuously. For instance, heart-beat of a specific person having any sort of cardiac disease. This can help doctor to fetch heart-beat value of that patient recorded at any specific time instant. Conversely, long names raise the problems to fit in Zigbee maximum payload of 127 bytes, so naming schemes consider this factor also. Additionally, hierarchical names are human-readable, thus, still there is need to design secured hierarchical compact naming scheme to provide original data in the case of privacy sensitive applications like smart-health. Furthermore, in this context, the work in [145] analyses the aspects of layer 2 communication in an NDN-based IoT. Findings indicate that L2 broadcasting has a severe negative impact on efficiency and reliability of content replication, which can be mitigated using a proper name-to-MAC-address mapping. Hence communication to groups should a layer 3 control and take advantage of the address mapping. Moreover, in [146] authors provide a system (i.e., that translate NDN names and MQTT topics) to show how these elements can be assembled to build a safety-critical surveillance environment for the IoT.

Moreover, lookup for length-varying names is expected to be complex. Therefore, it is quite stimulating and difficult to design such lookup system for IoT constraint-oriented devices[123]-[147].

Current literature investigated and proposed naming scheme for any single application, for instance in [22] and [102] ICN naming schemes are proposed for smart-home and VANETs respectively. Therefore, we stimulate ICN-IoT research community to put efforts to find and develop a naming scheme with carefully selected general, collective and public prefixes to cover (identify) and refer all IoT applications [119]-[110]. We are still looking for a general and appropriate naming scheme that can solve all identified constraints.

B. In-Network Caching

Though identified as the major beneficial feature of ICN for IoTs, ICN-IoT caching has received a lot of attention by research community. By employing ICN caching in IoTs can save network bandwidth, reduce latency to get data and improve battery life of IoT devices [75].

Mostly ICN-based caching schemes force to include freshness value of content while deciding about caching the content [80]-[81]-[82]. While content popularity has been included in caching decision in [85] but still there is need to explore popularity of content using simple method.

A lot of research has been conducted for caching placement strategies while most of research efforts suggest LRU as appropriate cache replacement strategy [76]-[82]-[84]-[148]-[149]. The work in [150] designs and thoroughly analyses a cooperative caching scheme that maximizes sleeping cycles and minimizes energy consumption of constrained IoT nodes. They show in theory and experiment that a clever replication strategy can indeed save significant resources while increasing the content availability throughout a wireless IoT system. Cache coherency protocols are almost completely missing from current literature and hold a lot of potential to be explored for IoTs.

Above all, a complete caching management system is still not present in current literature. Caching management system should address the responsibilities of IoT nodes about sharing constraints to ensure privacy and security of IoT applications and about the validity of contents in a node.

C. Content Routing and Information /Content Delivery

ICN-IoTs involves data routing and forwarding mechanisms when consumer node is far-away from producer node or indirectly connected in multi-hop fashion. Mostly ICN architectures support content naming while some research efforts in ICN-IoTs support naming IoT devices [100]. To provide routing for these two different types of names, either content name can be directly used in routing or device name can be resolved through Name Resolution System (NRS) to find requested content [147].

D. Mobility

We refer mobility to both producer and consumer mobile nodes. Most of the ICN architecture designs argue that consumer mobility is inherently supported while producer mobility is not completely specified. ICN mobile data consumer simply re-issue interest message and network forwards this interest towards nearest and reliable data provider or data cached node. However, for ICN-IoTs most of the nodes can act as providers/producers of information. In IoT applications like VANETs, vehicles act as information producer about the road condition for instance, information about accident, road construction, and can even operate as information provider when these vehicles cache data to forward to other vehicles nodes. Producer mobility [151] categorization is provided in [126], these four approaches (tracing and mapping mobile producer, data can be moved to a near stationary place or data can be regenerated form other mobile producers in that region) can be implemented for IoT scenarios. Also a proactive technique [152] can be investigated for IoTs environment. To cope with provider mobility in ICN, an initial draft is presented in [127] through simple and easy to maintain anchor-less approach. We argue that this approach should be explored and can become very beneficial in IoT constraint-oriented devices having limited resources.

E. Privacy and Security

A full of potential research area is privacy and security of both user requests and data in ICN-IoTs applications. Although ICN provides authentication and access control at content level but content requests are stored in ICN intermediate routers and can be tracked by attackers [153]. Thus to maintain privacy at router level between user and producer, privacy algorithms are required. Also it is still not standardized to decide whether intermediate routers will be present in ICN-IoTs applications or not [154]. Moreover, public key infrastructure (PKI) is very complex to implement for constraint oriented devices as it requires much power in the implementation of trust management and key generation [77]-[99]. Therefore, light cryptography and light hash function can be evaluated and hence modified for constraint-oriented devices. Keys generation and management that include both key revocation lists and key distribution processes are still need to explore further for IoTs applications. In addition, a significant research area is control access strategies in which user authentication, their corresponding access privileges, cache access and updates are needed to be investigated for IoTs applications. Moreover, security of sensitive information, spoofing and sniffing is highly needed to explore and address as highlighted in [30]. In [155] ICN-based safety is discussed in health care applications and can be explored for other IoT applications like smart home, smart grid and smart traffic.

In a nutshell, a complete mechanism ensuring both privacy and security for IoT data and applications is missing in current literature and therefore there is a strong need to design a holistic solution in this perspective.

F. Edge Computing (In-network Computation) and Cloud Computing

From IoTs perspective, in-network computation is a mechanism through which data collected from constraint-oriented sensors initially processed and later on, refined data is transmitted towards requested host. In-network computation is necessary to reduce the amount of produced data while lessening storage and high processing requirements. Other advantages of in-network computation include easy management of mobile nodes, less and refined cached data, simple data routing and forwarding and hence it can improve network-life, battery-life at the cost of simple and optimal in-network computation algorithms. In-network computation is the base for a new trend known as edge computing. As we mentioned earlier in Table III and Fig. 2 that cloud computing is the mainforce which is involved in IoT life cycle to process and manage IoT contents. As cloud computing seperates producer and consumer of information, which increases delay and bandwidth during the transmission and reception of information to central servers of cloud computing just for processing of data and management of information. Moreover, it poses many privacy concerns which can occur during the reception and transimssion of content to/from consumer/producer. Due to these disadvantages, a new paradigm with the name fog computing is introduced to shift computing and storage capabilities towards end node or edge node of the network. Due to involvement of edge nodes and edge routers, fog computing is also known as edge computing [156]. As edge computing need to cache data before its processing and in ICN-IoT, ICN enables IoT devices to cache data naturally. Thus in ICN-IoT caching with edge computing, IoT devices can also process the cached data. Moreover in ICN-IoT, it is encouraged to cache data near to end consumers (end nodes) which helps edge computing further. As a consequence, edge computing (in-network computation) becomes a key player for ICN-IoT caching. In IoT applications like virtual and augmented reality based games which require realtime behavior with almost zero-delay can benefit from edge computing [157]. A distributed edge computing mechanism divides the whole task among different devices of the network and ICN instance name function networking (NFN) can improve working of many ICN-IoT applications including smart-home and health, VANETs and smart grid [158]. This NFN further explored for IoTs and extended with scheduling algorithm [159]. Three resolution strategies are defined to support edge find or execute (EdgeFoX), Find-and-Execute (FaX) and Find-or-Pull-and-Execute (FoP)aX. These strategies can be applied to smart home or smart building [160]. Further, roles and addition of added nodes to perform in-network computation is needed to explore. Moreover, there is need to explore that how in-network computation will be performed in case of mobile nodes with and without caching.

Other way to perform ICN-IoT data processing and computation by employing cloud computing [161]. Clouds can share the burden of processing while providing high storage and can be used for calculating the analytics of any specific ICN-IoT application. For instance, high electricity usage can be calculated and can be seen in a any specific town of the city. Therefore, cloud assisted ICN-IoTs are needed to design that can, perform complex calculations, provide big storage and act as backup in case of mobile devices [162].

G. Content Discovery

In ICN, produced content is published by producer by placing corresponding name in nearest ICN-based router and it is stored in router to fulfill further consumer queries. In ICN-IoTs, consumer requests can be satisfied in two ways: (i) content is provided from nearest router, (ii) content is fetched directly from content producer. While in second case, consumer devices may need data with specific constraints like freshness [80]-[84]. To provide content accessibility in efficient way through ICN, packet formats must be specified and re-designed to cope such needs that could lead to easy content discovery and efficient delivery towards consumer. *Interest Message* and *Data Message* should be modified in order to support push type communication in ICN-IoTs [75]. For this, name-based aggregation can provide improved latency and efficient information lookup [100]. However, issues related to content discovery include the need to resolve: (i) How to name continuously produced contents to provide efficient look-up? (ii) How to manage content discovery efficiently in highly dynamic environments like VANETs? and (iii) How to map and search contents from named-devices corresponding to content requests efficiently?.

H. Quality of Service (QoS)

As ICN-IoTs have to drive highly heterogeneous and constraint-oriented devices, e.g., limited memory, limited battery life and specific processing unit. With these constraint-oriented devices, ICN-IoTs specific applications QoS needs ,e.g., low latency for VANETs, smart city and smart grid, better scalability and high reliability for smart health, smart grid, smart house and smart personal applications, should be satisfied and are not yet considered to be explored. Therefore, there is urgent need to design QoS-aware protocols to evaluate the performance of ICN-IoTs for latency, reliability, resource-consumption and scalability. ICN has much potential to improve delay and save bandwidth to satisfy different QoS requirements. ICN striking features in-network caching, any-cast, multi-cast, adaptability to mobile devices and dynamic environments and content security at network layer reduces much efforts that needs to be done with TCP/IP.

I. Business Strategies and Models

It is essential as well as critical to design business models for ICN-based IoTs because IoTs is known to be very advantageous and useful in our daily life. Therefore, business-strategy-makers are highly invited to put efforts to decide policies for ICN-based IoTs.

We identify some main questions that are needed to be explored and answered by research community from the perspective of major entities involved in the designing of these strategies. From consumer side, researchers need to investigate following questions: *What benefits will customers receive by sharing the data of their own servers, lets say, data from home server, to be cached?*, *How will privacy of a consumer be endured?* and *How much a consumer have to pay to upgrade to ICN-based IoTs solutions?*. Potential solutions for this can include, for instance, to provide quality data through caching, smart-home owners can get some extra free electricity or extra coaching to reduce their bills, smart-car-owners can avail free driving tips or road condition notifications in advance. From service-providers one need to look for these following questions: *How ICN-based IoTs will help to improve the QoS?*, *How it will assist to increase revenue growth?* and *What they would need to offer customers for caching the data?*. Most importantly, every country government need to participate to decide the extent of data sharing.

However, we are far beyond this phase of designing business models and therefore, business policy makers need to involve stakeholders, consumers and manufacturers to decide analytical consensus.

IX. Conclusions

We discussed and presented related literature of both new paradigms IoTs and ICN. Then, requirements and challenges to build a reliable and inter-operable communication network architecture for IoTs are presented. Through this paper, 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 summarized 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 literature into four categories of naming, caching, security and mobility, and presented in four different sections. Moreover, relevant operating systems and simulators for ICN-based IoTs are discussed in next section. In the end, we present identified research gaps that needs research community attention to build ICN-based network architecture for IoTs.

Acknowledgment

This research is supported by the Computer Engineering Department (CPED) of the University of Engineering and Technology (UET), Taxila, Pakistan under a Full-time research scholarship, and in close collaboration with both University of West London, UK and Waterford Institute of Technology, Ireland.

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

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

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

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

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

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

[17] (2016) Cyber-secure data and control cloud for power grids. [Online]. Available: http://cdax.eu/

[18] (2016) Greenicn architecture and applications of green information centric networking. [Online]. Available: http://www.greenicn.org/

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] J. Granjal, E. Monteiro, and J. Sa Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 3, pp. 1294–1312, 2015.

[26] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems for low-end devices in the internet of things: a survey," *IEEE Internet of Things Journal*, 2015.

[27] 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.

[28] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[29] 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.

[30] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 3, pp. 1441–1454, 2015.

[31] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 3, pp. 1473–1499, 2015.

[32] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *Communications Magazine, IEEE*, vol. 52, no. 11, pp. 122–129, 2014.

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

[34] 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.

[35] (2014) Iot6.eu researching ipv6 potential for internet of things. [Online]. Available: http://www.iot6.eu/

[36] 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

[37] ——. (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

[38] (2012) Ietf constrained restful environment (core) working group. [Online]. Available: https://datatracker.ietf.org/wg/core/charter/

[39] (2010) The constrained application protocol (coap). [Online]. Available: https://datatracker.ietf.org/doc/rfc7252/

[40] (2011) Ietf 6lowpan working group. [Online]. Available: https://tools.ietf.org/wg/6lowpan/charters

[41] (2008) Routing over low power and lossy networks (roll). [Online]. Available: https://datatracker.ietf.org/wg/roll/documents/

[42] (2009) Rpl: Ipv6 routing protocol for low-power and lossy networks. [Online]. Available: https://datatracker.ietf.org/doc/rfc6550/

[43] (2011) Ietf light-weight implementation guidance (lwig) working group. [Online]. Available: https://datatracker.ietf.org/wg/lwig/charter/

[44] (2015) Irtf thing-to-thing (t2trg) research group. [Online]. Available: https://datatracker.ietf.org/rg/t2trg/charter/

[45] (2017) Integrating objects to create new networked services. [Online]. Available: http://www.etsi.org/technologies-clusters/clusters/connecting-things

[46] (2017) Iot standards and protocols. [Online]. Available: https://www.postscapes.com/internet-of-things-protocols/

[47] 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.

[48] 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.

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

[50] 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.

[51] 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.

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

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

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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

[63] 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.

[64] 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.

[65] 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/.

[66] 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

[67] 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.

[68] (2012) Information-centric networking (icnrg). [Online]. Available: https://datatracker.ietf.org/rg/icnrg/about/

[69] 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

[70] 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

[71] 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

[72] 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.

[73] 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.

[74] 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

[75] 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.

[76] 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

[77] 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.

[78] 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.

[79] V. Sourlas, L. Tassiulas, I. Psaras, and G. Pavlou, "Information resilience through user-assisted caching in disruptive content-centric networks," in *Proc. IFIP Networking Conf. (IFIP Networking)*, May 2015, pp. 1–9.

[80] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer driven information freshness approach for content centric networking," in *Proc. IEEE Conf. Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 482–487.

[81] ——, "A case for icn usage in iot environments," in *Proc. IEEE Global Communications Conf*, Dec. 2014, pp. 2770–2775.

[82] 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

[83] Z. Li, J. C. Point, S. Ciftci, O. Eker, G. Mauri, M. Savi, and G. Verticale, "ICn based shared caching in future converged fixed and mobile network," in *Proc. IEEE 16th Int. Conf. High Performance Switching and Routing (HPSR)*, Jul. 2015, pp. 1–6.

[84] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in *Proc. Int Recent Advances in Internet of Things (RIoT) Conf*, Apr. 2015, pp. 1–6.

[85] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of internet-of-things data," in *Proc. IEEE Int. Conf. Communications (ICC)*, Jun. 2014, pp. 3185–3190.

[86] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, and K. Drira, "Cache coherence in machine-to-machine information centric networks," in *Proc. IEEE 40th Conf. Local Computer Networks (LCN)*, Oct. 2015, pp. 430–433.

[87] E. Baccelli, O. Hahm, M. Gunes, M. Wahlisch, and T. C. Schmidt, "Riot os: Towards an os for the internet of things," in *Proc. IEEE Conf. Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2013, pp. 79–80.

[88] Bonnmotion. [Online]. Available: https://sys.cs.uos.de/bonnmotion/index.shtml

[89] 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

[90] 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/

[91] 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.

[92] 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.

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

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

[95] 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.

[96] 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.

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

[98] 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.

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

[100] 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.

[101] 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.

[102] 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.

[103] ——, "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

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

[105] 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

[106] 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.

[107] 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.

[108] 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.

[109] 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.

[110] 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 (In Press)*, 2018.

[111] 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.

[112] 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.

[113] 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.

[114] 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.

[115] 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.

[116] 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.

[117] 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.

[118] T. Mick, R. Tourani, and S. Misra, "Laser: Lightweight authentication and secured routing for ndn iot in smart cities," *arXiv preprint arXiv:1703.08453*, 2017.

[119] 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

[120] M. Enguehard, R. E. Droms, and D. Rossi, "Slict: Secure localized information centric things." in *ICN*, 2016, pp. 255–260.

[121] J. Suarez, J. Quevedo, I. Vidal, D. Corujo, J. Garcia-Reinoso, and R. L. Aguiar, "A secure iot management architecture based on information-centric networking," *Journal of Network and Computer Applications*, vol. 63, pp. 190–204, 2016.

[122] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "A secure icn-iot architecture," in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 259–264.

[123] Y. Zhang, D. Raychadhuri, R. Ravindran, and G. Wang, "Icn based architecture for iot," *IRTF contribution, October*, 2013.

[124] C. Anastasiades, T. Braun, and V. A. Siris, "Information-centric networking in mobile and opportunistic networks," in *Wireless Networking for Moving Objects*. Springer, 2014, pp. 14–30.

[125] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, and K. Drira, "Producer mobility support in named data internet of things network," *Procedia Computer Science*, vol. 109, pp. 1067–1073, 2017.

[126] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A survey of mobility support in named data networking," in *Proceedings of the third Workshop on Name-Oriented Mobility: Architecture, Algorithms and Applications (NOM2016)*, 2016.

[127] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng, "Anchor-less producer mobility in icn," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 189–190.

[128] ——, "Map-me: Managing anchor-less producer mobility in information-centric networks," *arXiv preprint arXiv:1611.06785*, 2016.

[129] A. Compagno, X. Zeng, L. Muscariello, G. Carofiglio, and J. Augé, "Secure producer mobility in information-centric network," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 163–169.

[130] S. H. Ahmed, S. H. Bouk, and D. Kim, "Rufs: Robust forwarder selection in vehicular content-centric networks," *IEEE Communications Letters*, vol. 19, no. 9, pp. 1616–1619, Sep. 2015.

[131] S. H. Bouk, S. H. Ahmed, M. A. Yaqub, D. Kim, and M. Gerla, "Dpel: Dynamic pit entry lifetime in vehicular named data networks," *IEEE Communications Letters*, vol. 20, no. 2, pp. 336–339, Feb. 2016.

[132] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annual IEEE Int Local Computer Networks Conf*, Nov. 2004, pp. 455–462.

[133] Freertos - market leading rtos (real time operating system) for embedded systems with internet of things extensions. [Online]. Available: http://www.freertos.org

[134] P. Levis, "Experiences from a decade of tinyos development," in *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, 2012, pp. 207–220.

[135] Welcome!-openwsn - confluence. [Online]. Available: https://openwsn.atlassian.net/wiki

[136] Ns-3 based named data networking (ndn) simulator. [Online]. Available: http://ndnsim.net/2.1/

[137] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnsim: An highly scalable ccn simulator," in *Proc. IEEE Int. Conf. Communications (ICC)*, Jun. 2013, pp. 2309–2314.

[138] L. Saino, I. Psaras, and G. Pavlou, "Icarus: a caching simulator for information centric networking (ICN)," in *Proceedings of the Seventh International Conference on Simulation Tools and Techniques*. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST), 2014, pp. 66–75. [Online]. Available: http://dx.doi.org/10.4108/icst.simutools.2014.254630

[139] M. Tortelli, D. Rossi, G. Boggia, and L. A. Grieco, "Ccn simulators: analysis and cross-comparison," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 197–198.

[140] Get started with contiki, instant contiki and cooja. [Online]. Available: http://www.contiki-os.org/start.html#start-cooja.

[141] H. Will, K. Schleiser, and J. Schiller, "A real-time kernel for wireless sensor networks employed in rescue scenarios," in *Proc. IEEE 34th Conf. Local Computer Networks*, Oct. 2009, pp. 834–841.

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

[143] Ccnx packet format. [Online]. Available: https://www.ccnx.org/packet-format/

[144] Type-length-value (tlv) encoding. [Online]. Available: http://named-data.net/doc/ndn-tlv/tlv.html

[145] P. Kietzmann, C. Gündoğan, T. C. Schmidt, O. Hahm, and M. Wählisch, "The need for a name to mac address mapping in ndn: towards quantifying the resource gain," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 36–42.

[146] C. Gündogan, P. Kietzmann, T. C. Schmidt, M. Lenders, H. Petersen, M. Wählisch, M. Frey, and F. Shzu-Juraschek, "Information-centric networking for the industrial iot," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 214–215.

[147] 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

[148] T. Zhang, H. Fan, J. Loo, and D. Liu, "User preference aware caching deployment for device-to-device caching networks," *IEEE Systems Journal*, 2017.

[149] H. Fan, T. Zhang, J. Loo, and D. Liu, "Caching deployment algorithm based on user preference in device-to-device networks," 2017.

[150] O. Hahm, E. Baccelli, T. Schmidt, M. Wählisch, C. Adjih, and L. Massoulié, "Low-power internet of things with ndn & cooperative caching," in *ACM ICN 2017-4th ACM Conference on Information-Centric Networking*, 2017.

[151] X. Jiang, J. Bi, Y. Wang, P. Lin, and Z. Li, "A content provider mobility solution of named data networking," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2012, pp. 1–2.

[152] H. Ko, Y. Kim, D. Suh, and S. Pack, "A proactive content pushing scheme for provider mobility support in information centric networks," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 523–524.

[153] D. Saxenaa, V. Raychoudhurya, N. Surib, C. Beckerc, and J. Cao, "Named data networking: A survey," *Computer Science Review*, vol. 19, pp. 15–55, 2016.

[154] A. Lindgren, F. B. Abdesslem, B. Ahlgren, O. Schel, A. M. Malik *et al.*, "Design choices for the iot in information-centric networks," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 882–888.

[155] D. Saxena and V. Raychoudhury, "Design and verification of an ndn-based safety-critical application: A case study with smart healthcare," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[156] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[157] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the internet of things: a case study," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018.

[158] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin, "An information centric network for computing the distribution of computations," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 137–146.

[159] Q. Wang, B. Lee, N. Murray, and Y. Qiao, "Cs-man: Computation service management for iot in-network processing," in *2016 27th Irish Signals and Systems Conference (ISSC)*. IEEE, 2016, pp. 1–6.

[160] C. Scherb, D. Grewe, M. Wagner, and C. Tschudin, "Resolution strategies for networking the iot at the edge via named functions," in *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*. IEEE, 2018, pp. 1–6.

[161] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang, "Towards software defined icn based edge-cloud services," in *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*. IEEE, 2013, pp. 227–235.

[162] E. Borgia, R. Bruno, M. Conti, D. Mascitti, and A. Passarella, "Mobile edge clouds for information-centric iot services," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 422–428.