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

Sobia Arshad[ID], Muhammad Awais Azam[ID], Mubashir Husain Rehmani[ID], *Senior Member, IEEE*, and Jonathan Loo

Abstract—Information-centric networking (ICN) is being realized as a promising approach to accomplish the shortcomings of current Internet protocol-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, and 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 (IoT) like environments. IoT aims to connect anyone and/or anything at any time by any path on any place. From last decade, IoT attracts both industry and research communities. IoT is an emerging research field and still in its infancy. Thus, this paper presents the potential of ICN for IoT by providing state-of-the-art literature survey. 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 IoT with appropriate classification. Furthermore, we present operating systems and simulation tools for ICN-IoT. Finally, we provide important research challenges and issues faced by ICN for IoT.

Index Terms—Content centric networking (CCN), ICN-IoT caching schemes, ICN-IoT mobility schemes, ICN-IoT naming schemes, ICN-IoT security schemes, information-centric networking (ICN), Internet of Things (IoT), named data networking (NDN).

I. INTRODUCTION

A. Motivation and Background

INTERNET of Things (IoT) aims 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]. IoT 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 transmission control protocol/Internet protocol (TCP/IP) architecture while raising many important issues.

Among these issues (from the IoT device perspective), one is naming (and addressing) every IoT device [2], [3]. As Internet protocol version 4 (IPv4) addressing space is exhausted, Internet protocol version 6 (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]–[6]. Therefore, efficient naming and addressing schemes for billions of devices (and contents) are not ideally available in Internet protocol (IP)-architecture. Furthermore, every device has different constraints and specifications which raise another issue of heterogeneity. This is due to the fact that IoT comprises on devices which are heterogeneous in terms of processing power capability, size, memory, battery life and cost. Moreover, 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 unavailability. Therefore, solutions like in-network caching (which are required to make data available) are missing in naive IP-based networking. In addition, IoT 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 IoT applications, for instance, VANETs, MANETs, and smart transport require better mobility handling [8], [9].

On the other hand, from data perspective, most of the IoT 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 (WSNs), have specific purpose to harvest information at the large scale [10]. Every device has

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*

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 IoT, 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 fulfil IoT requirements. Moreover, besides above-mentioned requirements, IoT huge data put additional requirements like data dissemination and scalability on the underlying architecture. To fulfil all these needs of IoT, information-centric networking (ICN) (which is a promising 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 data oriented network architecture (DONA), content-centric networking (CCN) [11], publish–subscribe Internet technology (PURSUIT) [12], NetInf [13], CURLING [14], CONET [15], mobilityfirst (MF) [16], C-DAX [17], and green ICN [18]. Among these ICN-based architectures, DONA, scalable and adaptive Internet solutions (SAIL), content mediator architecture for content aware networks (COMET), CONVERGENCE, and CCN all are dirty-slate while MF, PURSUIT, and named data networking (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 (MM), 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 message authentication code (MAC) layer. CCN could be applied just

above MAC layer especially in WSN. Xylomenos *et al.* [21] and Amadeo *et al.* [22] argued 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 masks the need of associating content with the IP address instead by name. The actual content delivery still requires TCP/IP interface or direct MAC (layer 2) interface.

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. Furthermore, naming the contents can resolve the address space scarcity issue of IPv4 and can enable scalability in an efficient way and can also offer 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 [21]–[23]. Therefore, these are the reasons that in this paper we survey ICN-based naming, in-network caching, security, and mobility schemes which are explored for IoT. List of acronyms used in this paper is provided in Table I.

B. Review of Related Survey Articles

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

Exclusively IoT emphasized surveys have covered the IoT basics including building blocks and characteristics, enabling technologies, smart potential applications, projects, and related

TABLE II
IoT AND ICN RELATED SURVEY ARTICLES

Sr#	Reference(s)	Topics Covered	Publication Year
		IoTs Related Survey Articles	
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	2015
8.	[5]	enabling technologies, smart potential applications,	2014
9.	[2]	projects and related research challenges	2010
		ICN Related Survey Articles	
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	
1.	[34]	Briefly identify ICN for IoT and Future Directions	2016

research challenges in [2], [5], and [10]. Different eight research directions for IoT are listed down in [6]. Context awareness solutions for IoT are discussed in [27]. Middle-ware requirements and solutions are surveyed in [24]. IoT security issues and their corresponding solutions are outlined in [25]. In [28], specifically Sybil attacks in IoT are discussed along with their defense schemes. Moreover, classification of operating systems (OSs) for IoT is presented in [26]. List of survey paper for IoT 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, publish–subscribe Internet routing paradigm (PSIRP), and NetInf. Xylomenos *et al.* [21] described ICN concept, its features and extended the research of [20] by adding three more updated architectures named CONVERGENCE, CONET, and MF. Moreover, Fang *et al.* [32] focused on ICN energy efficient caching schemes on the basis of content placement, cache placement and request-to-cache routing. While Zhang *et al.* [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 survey 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 details. 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 OSs, simulators, and detail research challenges for ICN-IoT research community.

C. Contribution of This Survey Paper

We mainly aim to discuss ICN for IoT. 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 IoT. With such goals, to the best of our knowledge, it makes this paper pioneer and unique in this field. We make the following contributions in this paper.

1) We provide very brief overview of IoT architecture requirements and major ICN architectures with respect to their suitability for IoT in terms of naming, caching, security, and mobility handling schemes.
2) We summarize ICN-based architectures for IoT.
3) We provide comprehensive survey of ICN-based in-network caching techniques for IoT and classification of these schemes on the basis of role of content and node properties in ICN caching mechanisms for IoT.
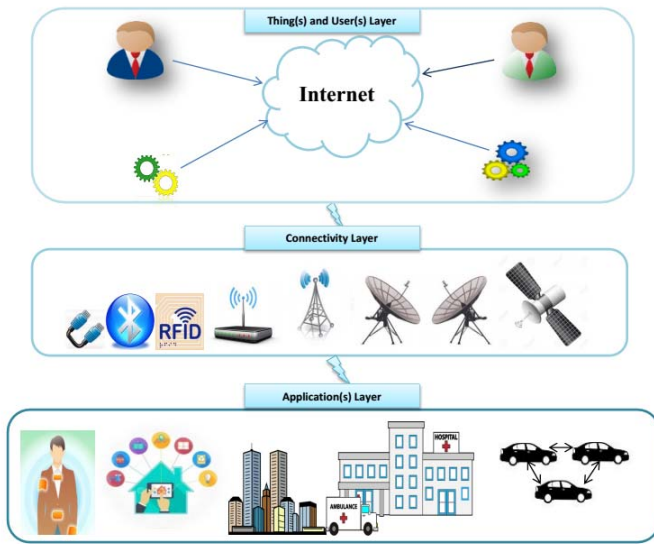
Fig. 1. IoT: connectivity types, Internet technologies, and IoT smart applications.

Fig. 2. Phases in IoT and corresponding enabling technologies.

TABLE III
IoT PHASES AND CORRESPONDING TECHNOLOGIES

IoT Phase		Components and Reference(s)
Acquisition and Sensing		RFID[45] WSN [27], [24] Bluetooth[46] NFC[47] UWB [48]
Data Transmission	Current Enabling Technologies	Ethernet[49] Wi-Fi[50], [51] Wi-MAX MANETs[52] Cellular Networks[53], [54], [55] Satellite Networks [56]
	Future Enabling (or Enabled by IoTs) Technologies	CRN[57] VANETs [58] 5G [59] ON[60] PLC[61]
Data Processing and Info. Management		Cloud Computing[62] Big Data[63]
Action and Utilization		Semantics[10], [64] Actuators[10] Applications[1], [10]

4) We provide classification of ICN-based content naming approaches on the basis of name structures for IoT.
5) We classify ICN-based security schemes for IoT on the basis of their security handling for IoT contents and IoT devices.
6) We categorize ICN-based mobility schemes into IoT producer mobility and hand-off management.
7) We classify famous ICN-IoT simulators and OSs and identify ndnSIM as a more explored tool for ICN-IoT.
8) We provide issues, challenges, and future research directions which ICN is facing for IoT.

D. Organization of This Paper

The rest of this 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 IoT. In Sections III–VI, ICN-based caching techniques, naming approaches, security, and mobility support are discussed, respectively. Section VII presents available OSs and simulators for ICN-IoT. In Section VIII, we present open challenges and future trends of ICN into IoT. Finally, Section IX concludes this paper.

II. INFORMATION-CENTRIC NETWORKING SUITABILITY FOR IoT

As IoT is the connectivity of things through the unified Internet. Therefore, things can be humans and smart machines of any sort and this is illustrated in the bottom portion of Fig. 1. These things can connect in three ways (connectivity in IoT can be seen in the top portion of Fig. 1).
1) Machine-type-communication.
2) Machine-to-human.
3) Human-to-human.
IoT works in four major steps.
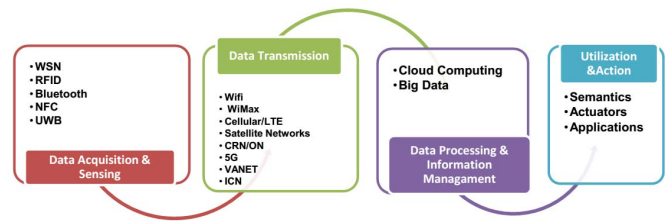1) Data acquisition or data sensing.
2) Data transmission.

3) Data processing and information management.
4) Action and 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 fulfils six purposes. First, we list and describe IoT architecture requirements. However, our aim is not to survey and discuss IoT in depth rather we illustrate it to highlight the related issues and identify architecture requirements. Second, we discuss IP-based evolutionary approaches for IoT. Third, we present the limitations of IP-based approaches. Fourth, we provide mapping of IoT requirements against ICN characteristics. In the next section, we describe briefly ICN-based proposed architectures with respect to their naming, caching, security, and mobility feasibility for IoT and lastly, we present some approaches which discuss and explore ICN for IoT.

A. IoT Architecture Requirements

Specific requirements and challenges [2], [5], [10] introduced by IoT network architecture outlined and given below.
1) Scalability: As IoT envisions not only connecting networks and corresponding devices but enabling low power devices in billions to connect through the Internet. Thus, it imposes new challenges over underlying architecture in terms of scalability. IoT architecture needs to support billions of devices in an efficient way. A current solution like IPv6 has a huge address space which can serve IoT devices. Although in future, addressing the IoT devices is not the only issue.

Another case is a large amount of data which is being produced by IoT devices, also needs better and efficient scalability management. Therefore, IoT network architecture must be explored in terms of scalability and it should be scalable to content access with network efficiency.

2) Mobility: Mobile devices like tablets, smart-phones have a small screen and limited battery life. Moreover, some IoT applications involve and require anytime and anywhere connectivity, in which users want to check their emails and/or make calls at anywhere and anytime. Furthermore, the number of mobile devices connecting to the Internet exceeds the stationary nodes. Therefore, to make data available at everywhere and provide fast and reliable connectivity, network architecture should support seamless mobility and roaming.

3) Security and Privacy: 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 the 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 a smart home where the detail of a pizza ordered by the house owner is required by a pizza shop to charge the payment. If this detail is shared with his doctor or insurance company, this can affect user privacy. As insurance company is not the tentative user and could use the private data in the wrong way. Therefore, privacy must be ensured via some access policies.

4) Naming and Addressing: IoT consists of billions of tiny, low-power, and constraint-oriented devices which need unique names or addresses to get recognition in the IoT network. For example, 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 in addressing and naming problem of IoT devices. But for constraint-oriented simple devices, it would be complex to process long address for very small communication because it results in the wastage of resources. On the other hand, IoT contents are being produced and processed at very fast speed. In addition, there can be many versions or values against any single content with different time stamps. Naming these rapidly produced IoT contents adds further complications. Thus, a larger and permanent naming scheme and addressing space are still highly needed for both IoT contents and devices.

5) Heterogeneity and Interoperability: It can be seen above that RFID tags and smart wireless sensors mainly build IoT. Smart sensors being major components of IoT offer many applications. These devices are heterogeneous in nature and usually vary in specifications like memory size, processing power, and battery life. Moreover, communication between these sensors is carried out by different underlying technologies (wired, wireless, cellular, Bluetooth, 4G, long-term evolution, CRN, and opportunistic networks). Thus, heterogeneous technologies are involved in communication. Therefore, IoT network architecture is required to support heterogeneity among device specifications and different underlying communication technologies and techniques in an interoperable way.

6) Data Availability: In the current TCP/IP-based architecture, whenever a node moves from one location to another, data which is assumed to provide, becomes unavailable. The 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 a time due to an occurrence of denial of service (DoS) attack. DoS occurs because the current Internet architecture cannot look at 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: Obviously billions of devices would need the huge amount of energy to build IoT 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 IoT.

B. Evolutionary TCP/IP Approaches for IoT

To attain the above-mentioned requirements and recent trends about IoT architecture, these have prompted many research organizations to initiate multiple projects. Therefore, many evolutionary (or dirty slate) approaches are being explored for IoT, for instance, IPv6-based 6LoWPANs [35]–[37].

Among these approaches, 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. The constrained application protocol (CoAP) [39] is a significant achievement that accomplished under CoRE working group. CoAP is a lighter version of the HTTP protocol. CoAP is designed mainly for low power devices forming constrained networks. CoAP also supports various caching forms which are mentioned in the representational state transfer protocol. CoAP runs over UDP to provide better communication among resource-oriented devices.

The IPv6 over low power wireless personal area networks working group (6LoWPANWG) [40] has focused on IPv6 over low power wireless personal area networks (6LoWPANs). This group works for adaption of IPv6 over IEEE 802.15.4-based networks. The 6LoWPAN group also works for IPv6 header compression to efficiently run over low power devices.

The routing over low power and lossy networks working group (ROLL) [41] mainly focus on developing routing strategies and self-configurable mechanisms in low power networks. Low power and lossy networks (LLN) made up of many embedded devices which include limited power and memory devices. LLN provides an end to end IP-based solution for routing over these networks. 6LoWPAN-WG will work closely to ROLL. Sometimes situations can happen in IoT when constraint-oriented devices are required to communicate with each other without any gateway. Therefore, IETF has designed the 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 [43] working group is focusing to build minimal and interoperable IP protocol stack for constraint-oriented IoT devices. The thing-2-thing research group (T2TRG) [44] aims to explore the factors which 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 with each other using machine-to-machine (M2M) communication style and with the global Internet.

Moreover, the European Telecommunications Standards Institute [65]] is working on the standardization of data security, management, processing and transport for IoT based on IPv6. However, more details about IoT projects and protocols can be found in [66]. Nonetheless, the above-mentioned projects for IoT architecture lies under "all-IP architectures" umbrella.

Furthermore, IP-based networking is inherently designed for host-to-host communication where location (e.g., address) of host plays a vital role, but this location-dependent design creates certain bottlenecks such as efficient information retrieval and delivery. Also, IP networking requires additional protocols to support privacy and security of sensitive data, scalability, mobility, and heterogeneity of nodes. Consequently, traditional IP-based networking is less suitable for these IoT devices and applications. Hence, to provide efficient connectivity among low power IoT devices, a novel networking model like ICN holds much potential [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 IoT

From both, today's Internet and IoT context, as all users need data even without knowing the producer of that data. More specifically, in IoT (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. Besides, flash crowds are also the apparent consequence of today's Internet usage [20], [21], [69]–[71]. Flash crowd is a situation which occurs on the Internet when a 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]. Moreover, data can become unavailable due to end of battery life of many sensors located in that producer vehicle. To minimize flash crowd, ICN provides and supports a much-needed characteristic named: *in-network caching* which minimizes traffic load on the 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 the original producer who
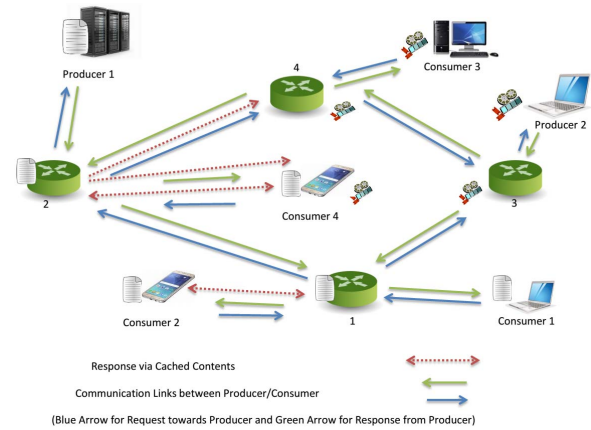

Fig. 3. ICN operation: consumer requests for a specific content by nearest routers (1–4) and producer replies and intermediate nodes caches that content and fulfils further request through cached contents rather than sending request toward the original producer.

cached that information item while reducing so-called flash crowd situation. 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 make ICN more suitable for IoT as it can combine billions of devices and huge information contents. As ICN supports *receiver-driven communication* making the communication under full control of the receiver. Therefore, IoT receiver of information which is more interested in data rather than its location can benefit from it. In addition, push type communication can be provided using beacon messages [73]. Furthermore, data can only be accessed whenever a receiver explicitly requests a data. As location-independent name searches data, so 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 regarding support for billions of IoT devices and the massive 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 billions of addresses and names to IoT devices and contents, respectively.

To support IoT applications which involve 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 reregistration after a hand-off of a mobile device with the nearest new router.

Security and privacy in IoT can be provided through following features of ICN. For example, ICN named contents make it easy to inspect that data is flowing according to query, content location independence hides the source of content,

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

receiver-driven communication style confirms that content arrives because the receiver has requested for this content and self-certified contents ensure that the contents are same as sent by the source.

Heterogeneity among IoT devices can be easily handled when devices named through ICN naming. Different types of IoT devices can operate with each other more efficiently when ICN strategy layer 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 IoT

This section presents naming, caching, security, and mobility support of nine famous ICN architectures such as DONA [74], NDN, COMET, PURSUIT, SAIL, CONVERGENCE, MF, C-DAX [17], and green ICN [18]. ICN major projects and architectures along with funding sources are presented in Fig. 4 and their feasibility with respect to 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 section, we present ICN-based IoT research efforts (in the 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, the research community is trying hard. In this context, NDN-based high level node architecture is proposed in [67] to support clean-slate architecture of ICN for IoT. Three layers NDN-IoT architecture which consists of an 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
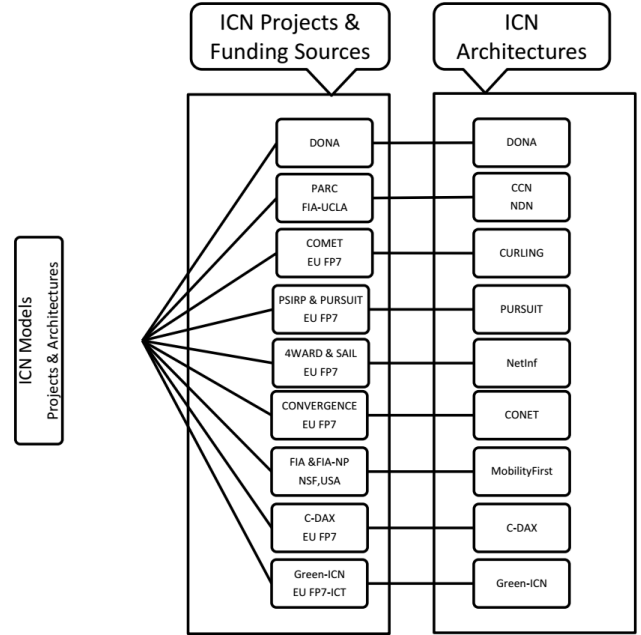

Fig. 4. ICN projects, funding sources, and architectures.

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 caching strategy. In Fig. 5, we present the evolution of Internet architectures. It shows the IP-based architecture, a 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 (IN), modifies interest message by including small data which is to be transmitted. This small data is not meant to be cached. Second scheme *Unsolicited data* transmits a small packet of uData that is not feasible for routing. In third scheme virtual interest polling (VIP), the receiver transmits long live interests such that whenever data is available, producer replies and on the failure consumer can retransmit interest message. They presented the analytical model for IN, unsolicited data, and

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, which 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
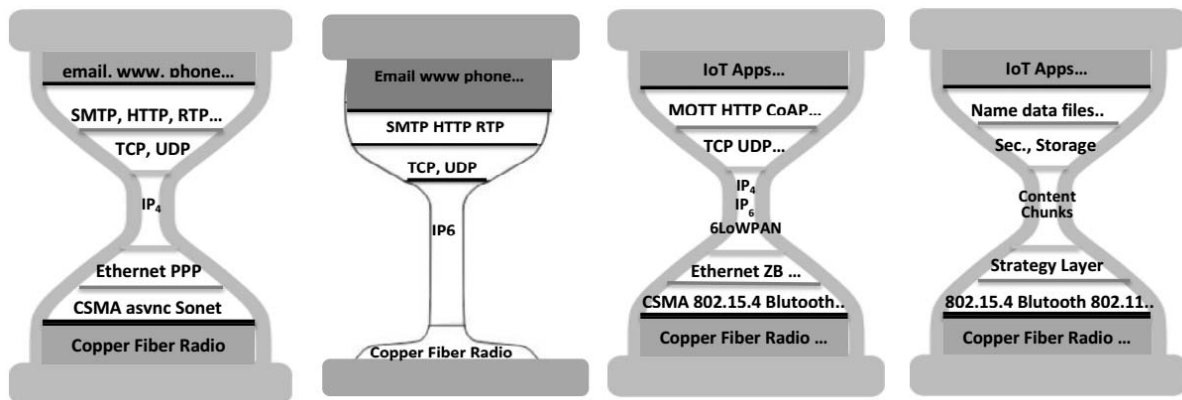
Fig. 5. IP-based network architectures and ICN-based IoT network architecture.

VIP and implemented the model in MATLAB. VIP outperformed in terms of used network resources and is suitable for massive IoT environment while the other two techniques are suitable for situations where the battery is a 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 implemented through 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 the 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. It is installed in the University of California at Los Angeles (UCLA) [77]. Name-based and encryption-based access control method is proposed and implemented to secure sensitive data. This is an 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, a unified ICN-based IoT platform is discussed in [78]. NDN and MF are selected to cater both static and mobile devices. They provided a comparison between 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 the 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 toward original producer which increases network load, retrieval delay, and bandwidth consumption. Moreover, the current Internet lacks support for data dissemination and fast retrieval of the content. These issues raised the need for in-network caching. To cope with these shortcomings of the current Internet architecture, Content Delivery Networks (CDNs) were introduced. By employing CDNs, caching is deployed as an overlay patch at the 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 essential 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 MF 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 toward edge devices in a cost-efficient way. As some IoT applications need fresh contents with some specific timing requirements. Moreover, mostly IoT contents are ephemeral in nature that need to replace with the newer versions, for instance, the temperature value of a room needs to be monitored and updated continuously. Furthermore, as IoT nodes are highly heterogeneous, which may differ in the

processing resources (i.e., constraint-oriented and powerful nodes) and IoT networks are a mixture of wired and wireless technologies.

In IoT, caching at intermediate devices or routers offers many benefits. As the receiver is dissociated from original producer, therefore by caching the contents, security improves and scalability of IoT network increases [67]. Energy efficiency of constraint-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 an environment like IoT. To design ICN-based caching for IoT, 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), the distance of a node from the producer (or/and consumer) and remaining memory. On the basis of these mentioned observations, we provide caching placement strategies into the 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 the distance of a node from producer or position/role in the network in caching decision lies in this category. ICN-based caching node architecture and cache coherency are also discussed in this category.

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

1) Content placement into the cache, in this phase cache space is allocated to contents on the basis of content and/or node. Content placement schemes include like cache each and everything (CEE, universal caching), and probabilistic caching.
2) Content replacement from the cache, in this second phase, when the 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 like least recently used (LRU) and least frequently used (LFU).
3) Cache coherency of contents in the cache, in this phase, the validity of contents which reside in the cache, is checked.

Caching performance measures include retrieval delay, hit ratio, network lifetime (how long network will exist in terms of connectivity), interest retransmissions (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 IoT in [80]–[84] as depicted in Fig. 6. In the following sections, we survey
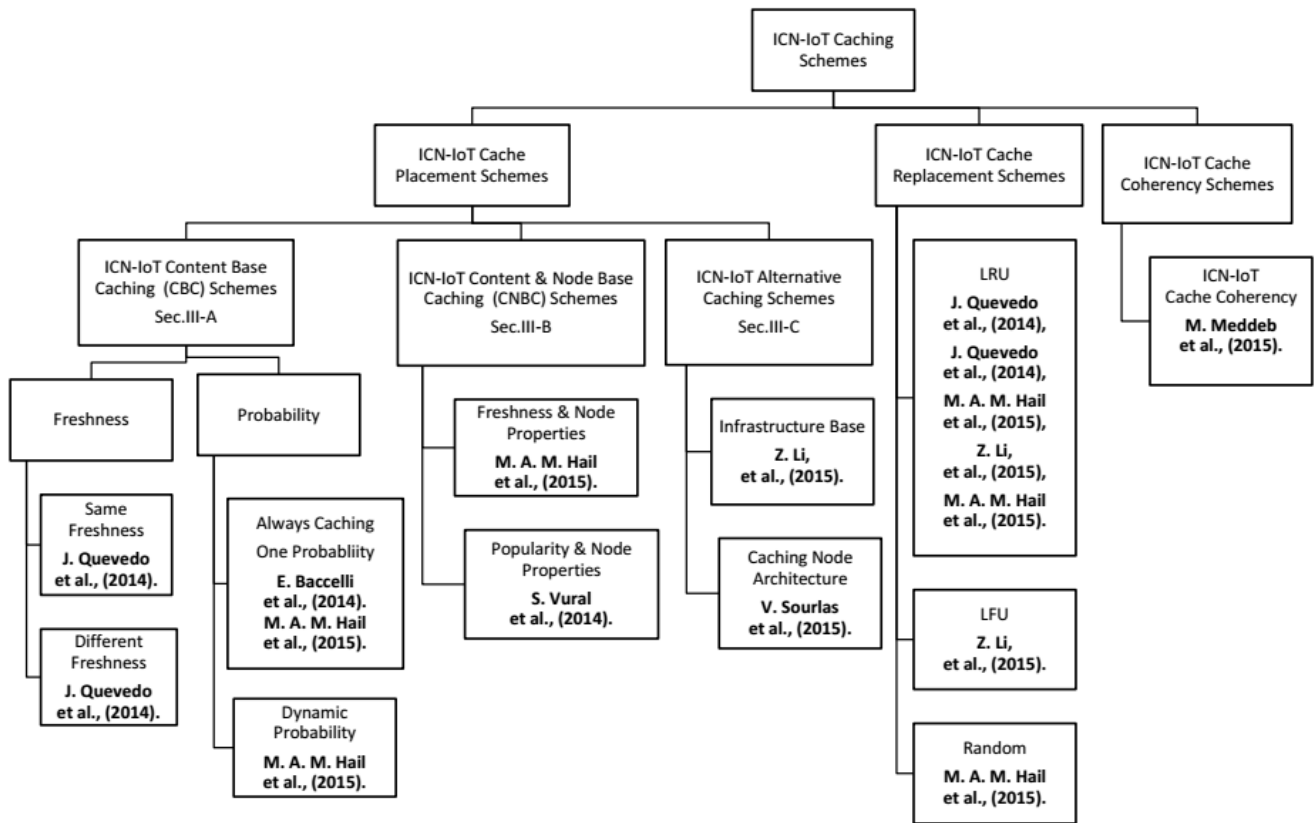
Fig. 6.   ICN-IoT in-network caching is illustrated in three phases: 1) caching placement; 2) replacement; and 3) coherency schemes. Caching placement schemes are further arranged into three categories: 1) CBC; 2) CNBC; and 3) alternative caching approaches.

caching placement schemes along with caching replacement schemes. According to Fig. 6, we subclassify caching placement schemes into three categories: 1) CBC; 2) CNBC; and 3) 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 subclassify alternative caching schemes into infrastructure-based caching, caching node architecture and cache coherency.

A. Content-Based Caching for ICN-IoT

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

1) Freshness of Content: IoT contents required by IoT applications are transient in nature which update their values continuously (e.g., temperature sensors update their values and consumer could request the most recent value or with 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 IoT.

In the following sections, we present those approaches which consider freshness in ICN-based caching design for IoT.

a) Specific freshness caching: In [80], a freshness-based caching scheme is proposed to facilitate consumer applications which inquire for contents with specific freshness values. The 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 content store (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. The consumer is assumed to send a request for the 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 (APs). LRU has been applied as a cache replacement strategy. Freshness value added more control of the consumer in the quality of data being fetched. By adding, a 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: IoT environment needs and corresponding ICN features are discussed in [81]. Bandwidth and energy consumption are measured for CCN-based IoT scenarios through varying number of nodes (both

TABLE VI
CACHING SCHEMES FOR ICN-BASED IoT ACCORDING TO THE CLASSIFICATION PRESENTED IN FIG. 6. CONTENT-BASED CACHING
IS FOR CBC 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 ++

consumers and producers) and compared against IP. CCN data packets are modified by including both freshness of content and fraction of size of CS. NS3-based ndnSIM have been used for IP and CCN, respectively. Application for the consumer is implemented in the way that it requests for the same data from different producers. Total one hundred nodes are included in the simulation while half of these nodes are producers and half were consumers. IP-based producers are assumed as WiFi mobile nodes connected to AP, while ICN consumer nodes are set to inquire data of same freshness value. LRU is applied as a cache replacement scheme and cache placement scheme has been designed to include freshness and variable CS size fraction. Impact of increasing sensors requires more bandwidth rather than the increasing number of consumers. This approach is good for IoT scenario where the number of consumers is 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, a small CS would be enough if freshness is highly required. However, considered IoT scenario has a fixed number of nodes and implementation is not performed for dynamic IoT scenario.

2) Probability of Content: Some IoT applications which require mix contents from the multiple or single producer(s) like in smart traffic, a car owner may be interested in the traffic condition ahead, the temperature of that area, the exact location of the vehicle and map toward its destination. Therefore,

ICN-based caching strategies for IoT should include factors to cope with these application requirements. In this context, random probability assignment can provide diversity in cached contents.

a) Always and probabilistic caching: Hail *et al.* [82] have implemented NDN for IoT 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 the simulation, out of which, four were destined as consumers and six were randomly selected as producers in a 400 m × 400 m area. Probabilistic caching scheme and LRU cache replacement scheme, in a combination, achieved higher results for the cache hit ratio, retrieval delay, and interest retransmissions. Cache size has been varied from 1 to 4 KB but optimal results were achieved when CS size was 4 KB. Probabilistic caching and LRU replacement scheme ensured content diversity and most recent contents in the IoT network which are important requirements of IoT. Though, authors have found caching (even with small CS) beneficial for IoT.

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 IoT because it saves bandwidth and energy consumption.

B. Content and Node-Based Caching for ICN-IoT

In this 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.

1) Probability of Freshness and Node Properties-Based Caching: Hail *et al.* [84] presented a probabilistic caching strategy for the IoT (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 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 a higher cache hit ratio and received data packets by the 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.

2) Popularity and Node Properties-Based Caching: In [85], a caching scheme has been proposed using data freshness, request rate, and router properties. Routers have 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 the node in the network). Numerical evaluation has been presented in MATLAB. However, the proposed caching scheme is for multimedia contents (40-GB link has been mentioned in simulation parameters) which 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), the 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 any caching replacement algorithm.

C. Alternative Caching Schemes for ICN-IoT

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

1) Overlay Caching for ICN-IoT: An overlay shared caching scheme based on ICN is presented in [83]. Content management (CM) layer is introduced in fixed and mobile converged (FMC) network architecture. This CM layer can be controlled through a 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 the network is responsible for transmission of content. A cache controller (CC) is integrated with UAG which provides optimal caching and prefetching 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 prefetching 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 a system is achieved in terms of less number of packets sent toward original server as more packets get a response from overlay caching, average latency, and uninterrupted playback than the current system. Presented caching strategy and management scheme offers caching as a service.

2) Client-Cache and Cache Coherency for ICN-IoT: 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 a client-based on-path caching strategy with less number of nodes and by using nodes that were close to the 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, leave copy down, probability caching, and 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 investigates cache coherency for ICN-based IoT. However, cache size that is selected, is much larger to suit for low memory devices to hold a large number of contents. Moreover, the discussion about IoT applications which require fresh contents is missing in the proposed method.

3) Caching Node Architecture for Disaster Management: Sourlas *et al.* [79] considered the disaster situation and presented the solution to recover data through cache enabled nodes. A caching scheme is presented to collect fragmented data when a network has 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 the producer is moved and network got fragmented (disruptive scenario). They tried to prolong a content availability through in-network caching because connectivity between friends and family is more crucial and as a result bulk of data is produced in such situations. NDN router architecture is modified by augmentation of SIT. SIT keeps track of users with the same interests and get the required data. SIT is meant to forwards 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 the producer or satisfied consumer by introducing distention flag (DF). If DF is 1, SIT provides the satisfied user with the same interest and now provides 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 node in IoT disaster management applications). Moreover, it requires other users willingness to disseminate data and respond to 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 IoT and provided a classification in Fig. 6. We have broadly categorized ICN-IoT caching mechanism into three phases: 1) caching placement; 2) replacement; and 3) coherency phases. Caching schemes have further categorized into three strategies: 1) CBC; 2) CNBC; and 3) 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 a few approaches. Therefore, ICN-based content popularity caching for IoT seeks urgent attention from the 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 caching mechanism, 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 the low processing power, memory, and battery. However, current literature on caching is missing IoT low power and low memory characteristics of nodes and IoT applications with mobile devices. Moreover, caching strategies lacks in push traffic type consideration for IoT network.

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

Besides both CBC and CNBC, we categorize remaining ICN-based caching schemes for IoT 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 irrespective 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 IoT application scenarios. On the other hand, CNBC schemes are somewhat challenging 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 common nodes due to its better results while LFU has been considered for edge nodes. Furthermore, 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 IoT [86]. Thus, ICN-based coherency protocols for IoT are highly required to provide content validation in IoT applications.

In a 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 IoT constraint-oriented nodes while accommodating both transient and ephemeral contents.

IV. ICN-IoT Naming Schemes

Fundamentally the IP-based Internet was designed to communicate between academic devices, but with time Internet usage has expanded from academic communication to fulfil society communication needs. Later on, with the help of add-on and specific purpose patches, IP-based Internet tried to fulfil current needs of society. As a consequence, by adding patches, IP-based Internet architecture provides current needs at the cost of more complex, slow, extra expensive communication, and sharing of content. With the time and keeping current expectations from the 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 a sender. The network is responsible for and will have to look for content providing best source [20], [21].

As users are more and more interested to receive content rather than the location of the content from where it is coming, so ICN approaches provide the ways to name data according to some constraints. The user can get desired contents by only providing their names.

ICN naming can also outperform in naming IoT contents. IoT contents are transient in nature and it is undoubtedly possible for one content to have many versions based on time and sensors which generate the same information. Moreover, IoT contents are huge in number like billions of 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 IoT smart and constraint-oriented connected devices
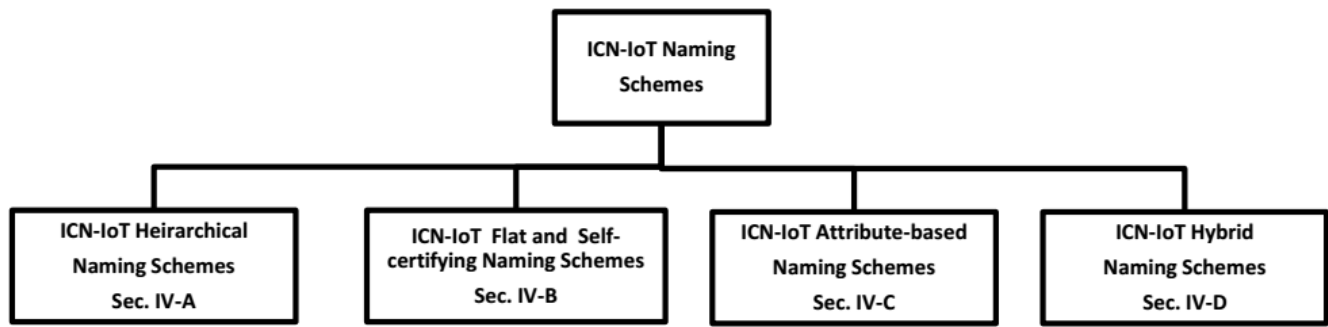
Fig. 7.   ICN-IoT naming is categorized into four categories: 1) ICN-IoT hierarchical naming schemes; 2) ICN-IoT flat and self-certifying naming schemes; 3) ICN-IoT attribute-based naming schemes; and 4) ICN-IoT hybrid naming schemes.

in 2020 [90]. Also, IoT network architecture is assumed to support scalability and heterogeneity.

Mainly there are two naming techniques (hierarchical naming structure and flat self-certifying/hash naming) which are available through ICN architectures. CCN [91]/NDN [92] name contents in hierarchical manner while other ICN approaches (DONA [74], PURSUIT [93], COMET [14], MF [16], SAIL [94], and CONVERGENCE [15]) follow flat self-certifying names. Third naming scheme, attribute-based has been used initially in combined broadcast and content-based (CBCB) 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 IoT [22], [34], [76], [77], [98]–[100]. Some researchers focus on hybrid naming schemes which incorporate both hierarchical and flat with attribute-based naming [101], [102]. We categorized ICN-IoT naming schemes into four types which can be visualized in Fig. 7 and summarized in Table VII.

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

A. Hierarchical-Based ICN-IoT Naming

These names are human-readable names and offer name aggregation. Moreover, NDN and CCN approaches use hierarchical naming. 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 which fulfils 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 also for low power devices, it could create more performance degradation.

In [100]–[103], hierarchical content naming scheme is used to name the contents. This paper is 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 a hierarchical name attributed to content and consists of a simple series of components of arbitrary lengths. No limitations are imposed that what sequence of the bytes 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 the prefix to provide multiple responses. They used CCN for every node. Contiki OS is used with Cooja simulator to simulate physical TelosB [104] 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, the 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 an 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 namespace 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 subtype (is chosen from real tasks like "/light, /temp, and /airCond"), respectively. Name aggregation is suggested to support task aggregation to reduce the number of sent messages and hence to reduce network bandwidth. However, they did not provide any simulations to show how names are carried by interest and data messages. The proposed naming scheme is designed for the home scenario and thus cannot be used for IoT applications which 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,"

TABLE VII
ICN-BASED IoT NAMING SCHEMES ARE SUMMARIZED ACCORDING TO 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					
[103]-[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
[77]	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					
[105]	CCNx	IP-based WSN	1.Average energy consumption 2.Average delay	WSN	Contiki OS and Cooja Simulator
[106]	ICN	Not provided	Not provided	Not for low-power IoT devices	No Simulations Not mentioned
Attribute-based Naming Schemes for ICN-IoT					
[107]	ICN	With and without ontology	1. Storage Overhead 2. Transfer Time Consumption	Smart Hospital	C Language
Hybrid Naming Schemes for ICN-IoT					
[101]	NDN	No Comparison	-	Vehicular Ad-hoc Networks	No Simulations Not mentioned
[108]	NDN	No naming Comparison	1.Start-up delay 2.Playback Freezing Ratio	Multimedia Contents dissemination in VANETs	NS3 with ndnSim
[102]	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
[109]	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

here "UetTaxila/CPED/VipLab/Light01/" represents light numbered as "01," located in Video and Image Processing Laboratory (VipLab), Computer Engineering Department (CPED), University of Engineering and 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.

Baccelli *et al.* [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 nonhuman-readable names are highly suitable for IoT smart devices while maintaining name-aggregation.

Further, Shang *et al.* [77] believed that hierarchical, human-readable, and application-specific names simplify both creation and processing tasks. NDN naming scheme is implemented to secure using ICN for UCLA campus. The designed prototype is implemented in Python and embedded in a browser-based interface. Namespace comprised of main root name followed by two subcategory 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 the specified time of Strathmore building located in UCLA. Moreover other sub-namespace, "/ndn/ucla.edu/bms/user/public/key/<key-id>" directs NDN-based building management system (BMS) application toward the public user (having multiple keys) through a user-specific key.

However, we argue that short but fixed 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], MF [16], and *NetInf* [13] follows the flat, short, and self-certifying names. These names can be computed using the hash of the content or any subcontent as part of it and thus can be nonhuman-readable. Moreover, flat names can be of any fixed length and therefore simple and easy to process in routing because these names take less computing resources and consume less space to cache.

Although there are very few research attempts which explored ICN flat naming alone. We survey and present these flat naming research efforts in the following paragraphs. Moreover, as such these efforts cannot be used for IoT but in hybrid form.

Dinh and Kim [105] presented ICN flat naming scheme for WSNs. The presented naming scheme has two parts: the first is to identify the category and the second is for content. They have investigated CCN naming in Contiki OS and results indicate that the proposed naming scheme outperform IP regarding energy consumption and delay. As this scheme is for WSN, therefore it can play an important part in IoT applications which involve low-power sensors.

Hong *et al.* [106] presented routing scheme based on flat naming. Bloom filter is used to provide name aggregation and efficient searching. They have introduced the concept of containers to save the contents. Controllers controlled containers and accessed through access controllers. Flat names play a great role in the routing of named contents because these are short in length which makes it easy to route and less complex in comparison. However, this paper has not involved constraints required by low-power constraint-oriented devices. Hence it is not suitable for all IoT applications.

Bari *et al.* [33] surveyed naming schemes of ICN architecture and argued that self-certifying names provide name-persistence, security-binding, and universal uniqueness. Moreover, [110] provide naming schemes comparison 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 [111], [112].

However, on the other hand, flat names does not provide name-aggregation which is needed for IoT contents and devices to ensure scalability. As flat names can increase the routing table entries making it more complex. Therefore, it will increase the delay in processing a query and will need ample space. Moreover, most of the flat names are nonhuman-readable, thus to respond any query, a third-party translation mechanism will be required. Furthermore, as 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 extracts attributes of the content. It is also used initially in CBCB [95]. As content attributes can include production date and time, content type, content location, content version number, and any specific property of the content. Therefore, this naming approach does not ensure global uniqueness of the content because it is possible to find many responses against a single query and it is hard to find unique content in a short time. However, it supports searching using easy and known keywords for the content. To secure contents, a routing scheme is provided in [113] using attributes of the content. In [107], an 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 provides better privacy, simple namespace management and reduces computation cost for the 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.

Bari *et al.* [33] believed and suggested to use keywords of content created by owner as they take less time in searching while making the lookup process easy.

For IoT applications, attribute-based naming can help in a perspective that IoT applications are extremely different and the user can specify the required content name in keywords. Attributes can be saved as keywords 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 fulfil this other naming schemes can be combined in a hybrid fashion.

D. Hybrid ICN-IoT Naming

Hybrid ICN-based naming schemes for IoT refer to naming schemes which combine three naming schemes or any two of them. The purpose of combining the 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 [101], a scalable naming scheme is proposed for mobile nodes like vehicles and their produced mobile contents. The content name consists of three components.

1) A scheme, "vhn" which specifies the vehicular network or vehicular identifier.
2) A prefix which is purely a hierarchical component and contains information of producer (car) and details about content.

3) The flat part is the hash of the item-name, owner-name, or signature of owner.

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

In [108], a hybrid naming scheme is proposed and used for multimedia contents in VANETs using ICN. Proposed naming scheme comprised of the following three parts.

1) Prefix "hmn": indicates "hierarchical multimedia naming" and hierarchical component names which also used for routing and name-aggregation.
2) The flat part is the hash computed on the complete name or part of it.
3) Attribute part is the attributes of the content.

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

Bouk *et al.* [102] investigated hybrid naming scheme proposed in [101] and presented their corresponding results for VANETs. Authors claimed that the proposed hybrid naming scheme take less space to save more names as compared to NLAPB [114] and simple trie. They have performed simulations, and results indicate that lookup time and memory management improves for VICN. Maximum prefix allowed length counted as 72 bytes. Therefore, this hybrid naming scheme is well suited for low power devices and can support IoT devices when underlying technology is IEEE 802.15.4 ZigBee (i.e., payload size is 127 bytes).

In [109], we propose a hybrid naming scheme for IoT-based smart campus. Hybrid naming scheme names the IoT contents while combining hierarchical and flat components. The proposed naming scheme takes the domain name, location, task as hierarchical component, and hash of device name as a flat component. The flat component is computed through FNV-1a hash to maintain the integrity of the content. The proposed scheme is evaluated and simulated for both static and mobile ZigBee devices in Contiki OS with Cooja simulator. Results show the better performance is achieved in terms of interest satisfaction rate, the number of covered hops, and name-aggregation ratio.

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

E. Summary and Insights

In this section, we have surveyed ICN-based naming schemes which are proposed and investigated for IoT applications. We categorized ICN-based naming schemes for IoT into four categories: 1) hierarchical; 2) flat self-certifying; 3) attribute-based; and 4) hybrid naming schemes.

Our survey indicates that for IoT, NDN (CCN) hierarchical naming schemes and hybrid naming schemes gained more attention from the research community as compared to flat and attribute-based naming schemes. We observe that the main reasons behind NDN (CCN) hierarchical naming feasibility for IoT are, 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 also name-aggregation saves a lot of space while making routing easy.

On the other hand, ICN-based hybrid naming schemes enhance the benefits of combined naming schemes. A 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 the flat-name component is concatenated to ensure improved security and privacy. Attributes of content are also 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 IoT. Although fixed length, nonhuman-readable flat naming provide better security and privacy through more easy and simple computations. However, this scheme does not provide better scalability, name-management, and aggregation. Moreover, this is the apparent cause behind less motivation to explore flat naming for IoT. Though, we highly suggest to use flat names to meet IoT 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 one and ten during the time 04:00 A.M. to 06:00 A.M. for any specific date from the desired area) which require contents according to specified features. Thus, we recommend that attribute-based naming should be explored for IoT.

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 the design perspective. As almost all IoT applications tend to take data from our daily life gadgets, then with the help of third parties that data is processed. The involvement of third parties creates a potential to affect our privacy. Moreover, content security is 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 extensible authentication protocol (EAP), PANA, SSL, DTLS, and IPv6-based security solutions employ the location of nodes. These security protocols secure communication channel between nodes rather than content. By adding security as a patch on IP, constraint-oriented IoT nodes perform their duties with more delays. Handling of mobile devices further complicates the situation. However, the IoT system is completely secured only
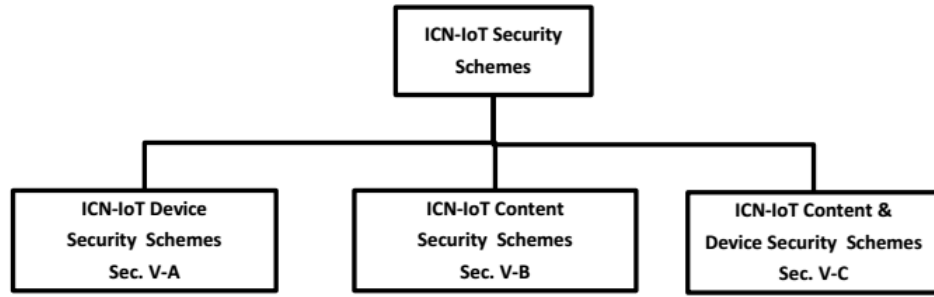
Fig. 8.   ICN-IoT security is categorized into three categories: 1) ICN-IoT device security schemes; 2) ICN-IoT content security schemes; and 3) ICN-IoT content and device security schemes.

when it ensures authentication, authorization, confidentiality, and integrity.

Moreover, ICN offers security at the network layer because it provides contents-based communication. Content-based security provides easy and straightforward security to IoT contents without the involvement of third-parties or external intermediate nodes. Content-based security maintains the content integrity and data authentication. Furthermore, ICN contents can specify content access control toward its users because ICN contents are generally known as self-certified contents.

We categorize ICN-IoT (ICN-based IoT) security schemes into the following three categories: 1) ICN-IoT device security schemes; 2) ICN-IoT content security schemes; and 3) ICN-IoT content and device security schemes. ICN-IoT device security schemes deal with device authorization and authentication, and ICN-IoT content security schemes provide content integrity and confidentiality. Next, ICN-IoT content and device security schemes provide security to both content and devices. Categorization in ICN-based security for IoT can be visualized in Fig. 8 and summarized in Table VIII.

A. ICN-IoT Device Security Schemes

In [115], an ICN-based secure protocol is proposed which provides security regarding both authentication and authorization of IoT devices. They call this ICN-based security protocol as on-boarding protocol (OnboardICNg). OnboardICNg protocol authenticates every joining device and authorizes 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 according to the ICN design for IoT. Through OnboardICNg, IoT network is secured from internal and outside adversaries. They compare OnboardICNg with preshared key EAP (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 useful for IoT 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.

Mick *et al.* [116] enhanced onboarding authentication protocol and combines routing with it. They called the proposed protocol lightweight authentication and secure routing (LASeR) protocol. They considered IoT smart cities made through islands. The considered scenario has anchor nodes, standard nodes, and gateway nodes. Among which only standard nodes are IoT nodes. An island manager (IM) just like AAM in [115] is used to authenticate and authorize the nodes. LASeR protocol works in three steps: 1) discovery phase; 2) authentication phase; and 3) advertisement phase. They evaluated LASeR in terms of convergence time and transmission burden for the different number of nodes and increasing distances among nodes. The LASeR only focuses on authentication with routing. However, IoT nodes do not involve in this whole procedure, and they delegate their duties to anchor nodes and IM. Like [115] they also talk about securing the IoT applications and nodes as a whole.

B. ICN-IoT Content Security Schemes

Bouk *et al.* [102] have presented secured content naming scheme where a content name is secured using Base64 format. This paper only considers multimedia contents fetched by vehicles. The secured part is included at the end of interest packet and can be calculated by taking the hash of attributes of content or a public key of the vehicle. They have programmed it in Linux-based C++ programming. They have only considered vehicles and not static devices.

In [117], we propose an 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 subtype name is encrypted through SHA256. Moreover, the name of the node which is originating the interest is also encrypted through SHA256. Security is preserved in the context of integrity. However, no implementation results are presented.

C. ICN-IoT Content and Device Security Schemes

NDN-based architecture to secure any building is presented in [77] and installed in UCLA. It is 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

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

Ref.	Model	Security Perspective	Methodology	Comparison	Parameters Evaluated	Finding(s)	Simulator (OS, Programming Platform, Language)
ICN-IoT Device Security Schemes							
[115]	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
[116]	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							
[102]	CCN	Integrity	Base64 Format on Content name	No Comparison	No Implementation	Maintains Integrity of content name and device name	Linux-based C++ programming language
[117]	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							
[77]	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
[118]	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
[119]	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
[120]	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

sensor devices run IP-based BMS protocols. Manager application is controlled by a human operator who authorizes out of band users. Manager application is also responsible for NDN management and auto-configuration of sensors and gateways. Gateways publish contents into NDN repositories. NDN repositories are responsible for responding user queries about sensors data. In NDN-based BMS, they followed and designed hierarchical naming to name devices and contents. They used public keys of the user and append it as the last component of content name by calculating its hash through SHA256. Two lists are maintained to maintain user privileges. Each gateway has an access control list (ACL), which is a list of identities of authorized users. Another list, access privilege list (APL) which is maintained by every user of BMS, contains the data namespaces that any user can access. APL is also published in NDN repositories. To provide the mapping between content namespaces and user IDs both lists (i.e., ACL and APL) are responsible which 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 the user gets capability-certificate to access data. During gateway configuration, NDN packets are signed and encrypted using the symmetric key to secure from man-in-middle attacks. Sensor data is encrypted through the shared symmetric key to providing access-control and published in JSON format. Gateway generates and distributes symmetric keys while going through ACL. It publishes encrypted key (encrypted through the user's public key) asymmetrically. Data packet also contains time-stamp of the decryption key to ensure content-based security. Python-based data publishing service is used to publish data and browser-based data visualization application. The data publishing service packs data in JSON format into NDN repositories. The user issues interests using data visualization application and can employ time-stamp filter. The user gets encrypted data and decryption

is performed through the 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 of sensor uploads data into NDN repositories first. However, it enables caching, lowers load on the data server, and preserves IoT scalability as data is secured via encryption only single time.

Enguehard *et al.* [118] discussed forwarding and security for ICN-based IoT. Geographic forwarding is implemented due to its low control traffic for sending data toward the destination. It involves the 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 the beacon, contain MACs generated through encryption using broadcast keys. However, if the neighboring node is tempered then this scheme is not resilient. They evaluate their proposal in RIOT OS in terms of computation, network and memory footprints. It takes 28–35 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, and it is one order lower than the 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 IoT through securing IoT devices and contents.

Suarez *et al.* [119] discussed benefits and challenges of applying ICN for IoT. They considered two content requests: 1) when any user wants an action performed by any device and 2) when the user requests the current content of the device. Their proposal consists of a gateway, admin, clients with the 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 a 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 the discovery procedure, the client discovers a list of IoT devices. In its

working, as step 1, the client first expresses an interest in the form of CCN name. In step 2, the gateway receives this interest and respond with the data packet. Data packet indicates content protection and also provide information to the 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 a cryptographic digest of its public key and key locator is responsible for the actual location of the public key. In step 3, in order to get appropriate key, the client issues an interest in the protection of exchange information. Then, the client gets verified through the gateway to enable IoT service routine. When the client is authenticated, the gateway generates a random systematic key SKcg (128-bit AES key) for cryptographic functions. This SKcg key along with its related information is encrypted with the public key of the client as extracted from the data packet in step 4. Data provided by the gateway is verified and decrypted by the client through its SKcg. The client also generates a message authentication code (MAC) over the whole interest by using the session key SKcg. In step 5, MAC and a unique nonce value are appended with CCN name to prevent malicious attacks. Gateway verifies nonce and MAC component and replies to interest message with the data packet in step 6. As step 7, the client can retrieve information from the gateway by issuing interest after validation of client. Then gateway reply client in accordance with client specific policy in step 8. The gateway-based proposed design presented in this paper have the flexibility to adopt according to environment and organization. It also enables security feature through the built-in support of automatic discovery and registration process which is the uniqueness of this design. It also reduces the overall incoming interest packets. The result shows that the average service time of interests is stable for 25 requests per second. This paper is suitable for IoT as it can scale up with less overhead and secures both IoT contents and devices.

Sicari *et al.* [120] proposed an ICN-based secure architecture for IoT. The proposed ICN-IoT secure architecture provides a 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 integrated security with ICN-IoT architecture [121] 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 sends its device ID, signature key and certificate; this triplet is sent toward aggregator where it verifies and stores this new device information. Then aggregator issues an action key encrypted through the signature key. If the new joining device is not a certified device, then it can send its device ID only. In this case, the aggregator can issue a 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, the user further sends its actual query/request in encrypted form through its action key and signature key. ICN-IoT server forwards this request toward aggregator. Aggregator decrypts and satisfies the request with the help of IoT nodes and sends relevant response toward corresponding IoT user. Secure naming service provides security to names of IoT devices. Aggregator sends device ID, a signature key and action keys toward LSG which in turn assigns the name to device and replies name to the aggregator. Aggregator sends device name toward device by encrypting it through action key of the device. During a subscription, a user needs a secure subscription which 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 the device is ensured by sending device name, ID encrypted with signature key and data encrypted with action key to the aggregator. Aggregator decrypts data and sends to ICN-IoT server. ICN-IoT server again encrypts data with the action key of the user and sends toward the user. Proposed ICN-IoT architecture aims to secure both content and device by maintaining privacy, authentication, confidentiality, and integrity. However, the authors did not 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 the first category, we listed and summarized those approaches which handle the ICN-based security of IoT devices. These approaches mainly provide authentication and authorization of IoT devices. The second category, ICN-IoT content-based security schemes mainly deal with content and aimed to provide content integrity, nonrepudiation, and confidentiality. The resulting contents are self-certified which can specify its owner details and content details. In the 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 on securing the whole IoT system while providing content integrity, confidentiality and device authentication and authorization. Moreover, some techniques also consider access-control-management which aims to specify the list of intended users.

Our survey finds that ICN-based security schemes must be designed which involve IoT environment characteristics; for example, 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. However, most methods which are discussed above, apply security methods over interest and data messages. Therefore, there is a need to ensure that security mechanisms must provide authenticated requests along with push support enabled.

Moreover, public key cryptography (asymmetric cryptography) cannot 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 embed security information at the end of query/interest packets as last named component, result in 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 an important part and is explored in many approaches like [77], [99], [115], and [116]. 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 also make symmetric key cryptography inflexible for IoT.

Besides these, nowadays elliptic curve cryptography (ECC) is being explored for IoT constraint-oriented devices because of its simplicity and extra lightweight nature. ECC utilizes elliptic curve theory to produce better cryptographic keys in terms of size and efficiency. As compared to the 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 and 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 no single solution fulfils all requirements of IoT nodes and applications. Therefore, ICN-based IoT security solutions must be designed in a flexible way which includes both IoT application requirements and device specifications and capabilities.

VI. ICN-IoT MOBILITY SCHEMES

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

Although there are other mobility models (like nomadic and pervasive), however, in IoT, cellular mobility model plays an important role. In cellular mobility, wireless networks are divided into cells and each cell has specific radius and area of service. When mobile devices move from one cell to the next, they face a situation called a handoff condition. Therefore, handoff-management also becomes an essential factor to solve.

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							
[122]	NDN	Producer Mobility	Content transfer content discovery	No Comparison	No Implementation	Delegating content retrieval to agents is better	CCNx
[123]	NDN	Producer Mobility	Survey	No Comparison	delivery cost path length interest routing	Name-based routing is better	Analytical Evaluation
[124]	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							
[125]-[126]	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
[127]	NDN	Secure Producer Mobility	Hash and Hash chains	MD-1,-5, SHA256, DSA, RSA	Computation Overhead Storage overhead	Lightweight attestation & Scalable	ndnSIM

In ICN-based IoT, 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. Therefore, in this way, the mobile subscriber can reissue interests for which they did not receive data. To support mobility, DTN function does not 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 MM schemes and other ICN-IoT mobility schemes. First category schemes combines those schemes in which ICN-based producer mobility is discussed and are summarized in Table IX. 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. First, producer location is needed to find and trach along with easy session maintenance. Then it is identified 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 the new location.

In [122], producer mobility support mechanisms and their disadvantages are discussed in three categories. Routing-based producer mobility is provided by updating the routing tables which involve the forwarding of information queries. However, the routing-based approach is not suitable to provide scalability of routing tables. Second, the indirection approach requires some extra nodes (home-agents) which keep track

of nodes locations and forward interests to the updated location of the 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). Also, every query and data message also visit this home-agent. The third approach, resolution-based include content updated location (or information about updated location) in data message as a response of user query. Resolution-based approach incurs the overhead of this one extra packet. They further discuss both content discovery and transfer mechanisms. This paper discusses 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.

In [123], NDN-based producer mobility is discussed for IoT. They discussed NDN-based producer mobility support through four approaches. The first approach solves producer mobility by utilizing the location information through the location resolution system (LRS). Producer updates LRS about its location after moving. LRS keeps the record of content name prefix and its corresponding producer. Consumer requests the location of the content producer by sending the message having content prefix toward LRS. In the second triangular approach, interest message is sent toward the previous location. Then using FIB update, it is rerouted toward the new location. The data message is delivered first toward old location and then from there, it is forwarded to the consumer. In the third locator/identifier separation approach, every content is managed in two parts by its producer. Content first part is its identifier and the second is its locator. In identifier, prefix or content name is stored and in the locator, the 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. The fourth approach, routing-based approach finds the data through the name-based routing protocol. Name-based routing protocol

tries to find the cached copies of data toward the path of the original producer. Name-based routing can be implemented through decentralized routing using flooding and distance-based greedy routing protocol. Thus, its complexity depends on the routing protocol. They expect that name-based routing scheme can perform better in IoT due to its average cost for packet delivery, less handover latency, and optimal routing patch length. However, they did not propose any technique for NDN-IoT producer mobility.

Zhang *et al.* [124] surveyed producer mobility and categorized into four categories: 1) mobile producer (MP) mapping; 2) MP tracing; 3) data depot; and 4) 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 toward MP. In MP tracing, interest messages can use traces of MP on the way toward RV and get forwarded toward MP without involving RV. In the data depot, a fixed 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 the data spot, new MPs generate data in order to fulfil 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 often and traces can provide a direct path toward MP.

1) Anchor-Less Producer Mobility: In [125], proposed producer MM scheme is designed to meet 5G requirements of low latency, low network overhead and overall fast speed. MM schemes are categorized into three classes: 1) anchor-based; 2) anchor-less; and 3) 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 anchor-based approach, a specified node is responsible for all nodes movements and direct messages to the new locations of moved nodes. They have proposed the anchor-less MM system to support delay-sensitive applications like smart health. When a patient is moving and acts as a mobile producer, its fast MM is important. They used stateful forwarding, ICN in-network caching and defined forwarding mechanism, to update and populate temporary FIB (TFIB) from producer new location toward 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 eliminates the need for in-network anchors while limiting the MM toward 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 provide necessary support. Anchor-less producer mobility is ensured in three simple following steps. Every mobile producer updates content (it produces) as a list of prefixes to its new PoA after establishing a 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 toward producer using this TFIB information or

using FIB along with discovery mechanism. In [126], 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 IN. Due to 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 delays sensitive applications like smart health.

Compagno *et al.* [127] identified loop-holes of [126] and proposed a prefix attestation protocol to secure trace-based producer mobility. Protocol map-me can be compromised when IU came from an attacker. It can pollute cache and disturb the 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, the producer sends minimal security context toward the registration server to generate valid IU. This security context is distributed 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 the attestation protocol analytically in terms of goodput. Goodput decreases when because IUs take resources. Hash chains maintain optimal goodput in case of one hash or multiple hashes per IU verification. Around 50 MB is 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 [128], a forwarding mechanism is presented for vehicles by incorporating one immediate vehicle resources. It ranks the vehicle upon multiple factors and selects one as forwarder among all vehicles. However, it does not account for provider mobility (i.e., adhesive issue of ICN mobility). Moreover, Bouk *et al.* [129] provided a scheme dynamic PIT entry lifetime to reduce the number of PIT entries. Therefore, it minimizes the usage of the battery of mobile nodes and makes routing and forwarding easy and fast.

C. Summary and Insights

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

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 MM and handling with less number of packets. Therefore, in this context, anchor-less producer

TABLE X
ICN-IoT OS AND SIMULATION TOOLS ANALYSIS

	ndnSIM [134]	Contiki OS/Cooja Simulator [26], [130]	RIOT OS [87]
Ref. #	[80]-[81]-[82]-[84]-[108]-[116]-[126]	[100]-[103]-[109]	[76]-[118]
Total # of Ref.	7	3	2

MM [125], [126] can be employed for IoT environment and can be secured further through the hash chains method presented in [127].

Moreover, in other ICN-IoT schemes, those schemes are included which try to make IoT mobile node lighter while minimizing PIT entries and selects the best forwarder among available vehicles.

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

VII. ICN-IoT OPERATING SYSTEMS AND SIMULATION TOOLS

There are a lot of IoT OSs and simulation tools which can be used for ICN-IoT. In [26], famous IoT OSs including Contiki [130], FreeRTOS [131], RIOT [87], TinyOS [132], and OpenWSN [133] are presented under the category of open-source and closed-source (which are not available commercially). Among these, we only discuss which can be used for both IoT as well as ICN implementations. On the other hand, specific ICN simulators (ndnSim [134], ccnSim [135], and Icarus [136]) are presented in [137]. 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], [130] is an open source and flexible OS developed at the Swedish Institute of Computer Science in Sweden. It is a very lightweight OS for sensor nodes which are severely resource-constrained regarding power, memory, processing power, and communication bandwidth. Contiki is developed in C language and is event driven. Main features of Contiki OS include the support of preemptive multithreading per-process and dynamic loading and unloading of code at runtime. A Contiki configuration consumes 40 kB of ROM and 2 kB of RAM. The communication between different processes always goes only using the kernel of the OS. A full installation of Contiki OS 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 [138] 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 Lesser General Public License and an open-source OS for sensor nodes in the IoT. RIOT OS is a microkernel-based OS inherited from fire kernel [139] which matches 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 have the ability to deal with the various challenges in powering of constrained devices networks. RIOT also provides both real-time capabilities and full multithreading. RIOT provides the C and C++ programming language supports for applications.

C. Other Simulators

NDN architecture can be simulated using its own specific ndnSim simulator. This ndnSim [134] is an NS3-based simulator and provide simulation for NDN and CCN.

Mini-CCNx [140] is a tool for agile prototyping of ICN-based on the CCN model. Mini-CCNx is used to build several CCN topologies, each with hundreds of nodes, great agility, and flexibility. These topologies can be run directly on laptop/desktop, in a local VM or cloud. Moreover, the best is: the code which user runs on Mini-CCNx is the same code used in a real network. This feature adds a realistic behavior to simulation tests. Each Mini-CCNx node (host or router) runs the official project CCNx's; therefore, the user uses the official CCN implementation.

ICN simulator is 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 massive amount of information, providing an insight on the new techniques introduced in the topology management of the information-centric network.

Icarus [136] is a caching simulator which supports multiple caching schemes and replacement schemes. It is a Python-based 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-IoT

In this section, we present issues with the current solutions for ICN-IoT 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) [141]. 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]–[142] does not have fixed length header. For small data packets (like mostly IoT applications have short length data to transmit in a response of a query or to send command toward 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 IoT applications [142].

In addition, as CCN/NDN naming follows the 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 the doctor to fetch the heartbeat 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 a 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 [143] analyses the aspects of layer two 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 three control and take advantage of the address mapping. Moreover, in [144] 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 a lookup system for IoT constraint-oriented devices [121]–[145].

Current literature investigated and proposed naming scheme for any single application, for instance in [22] and [101] 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 [109]–[117]. 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 IoT, ICN-IoT caching has received a lot of attention from the research community. By employing ICN caching in IoT can save network bandwidth, reduce latency to get data and improve the 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]–[82]. While content popularity has been included in caching decision in [85] but still there is a need to explore the popularity of content using a simple method.

A lot of research has been conducted for caching placement strategies while most of the research efforts suggest LRU as appropriate cache replacement strategy [76], [82], [84], [146], [147]. The work in [148] 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 smart replication strategy can indeed save significant resources while increasing the content availability throughout a wireless IoT system. Cache coherency protocols are almost entirely missing from current literature and hold a lot of potential to be explored for IoT.

Above all, a complete caching management system is still not present in the 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-IoT involves data routing and forwarding mechanisms when consumer node is far-away from producer node or indirectly connected in the multihop fashion. Mostly ICN architectures support content naming while some research efforts in ICN-IoT 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 [145].

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 reissue interest message and network forwards this interest toward nearest and reliable data provider or data cached node. However, for ICN-IoT 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 the accident, road construction, and can even operate as information provider when these vehicles cache data to forward to other vehicles nodes. Producer mobility [149] categorization is provided in [124], these four approaches (tracing and mapping mobile producer, data can be moved to a near stationary place or data can be regenerated from other mobile producers in that region) can be implemented for IoT scenarios. Also a proactive technique [150] can be investigated for IoT environment. To cope with provider mobility in ICN, an initial draft is presented in [125] 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 the potential research area is privacy and security of both user requests and data in ICN-IoT 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 [151]. Thus to maintain privacy at the 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-IoT applications or not [152]. Moreover, public key infrastructure 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 needed to explore further for IoT 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 IoT applications. Moreover, security of sensitive information, spoofing and sniffing is highly needed to explore and address as highlighted in [30]. In [153], ICN-based safety is discussed in healthcare 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 IoT perspective, in-network computation is a mechanism through which data collected from constraint-oriented sensors initially processed and later on, refined data is transmitted toward the 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, and 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 main force which is involved in IoT life cycle to process and manage IoT contents. As cloud computing separates 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 transmission 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 toward the end node or edge node of the network. Due to the involvement of edge nodes and edge routers, fog computing is also known as edge computing [154]. 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 real-time behavior with almost zero-delay can benefit from edge computing [155]. A distributed edge computing mechanism divides the whole task among different devices of the network and ICN instance name function networking (NFN) can improve the working of many ICN-IoT applications including smart-home and health, VANETs, and smart grid [156]. This NFN further explored for IoT and extended with scheduling algorithm [157]. Three resolution strategies are defined to support edge find or execute, find-and-execute, and find-or-pull-and-execute. These strategies can be applied to a smart home or smart building [158]. Further, roles and addition of added nodes to perform the in-network computation is needed to explore. Moreover, there is a need to explore that how in-network computation will be performed in case of mobile nodes with and without caching.

Another way to perform ICN-IoT data processing and computation by employing cloud computing [159]. 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 any specific town of the city. Therefore, cloud-assisted ICN-IoT are needed to design that can, perform complex calculations, provide big storage, and act as the backup in case of mobile devices [160].

G. Content Discovery

In ICN, produced content is published by the producer by placing the corresponding name in the nearest ICN-based router and it is stored in the router to fulfil further consumer queries. In ICN-IoT, consumer requests can be satisfied in two ways: 1) content is provided from the nearest router and 2) content is fetched directly from the 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 redesigned to cope such needs that could lead to easy content discovery and efficient delivery toward the consumer. *Interest Message* and *Data Message* should be modified in order to support push-type communication in ICN-IoT [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.

1) How to name continuously produced contents to provide efficient look-up?
2) How to manage content discovery efficiently in highly dynamic environments like VANETs?
3) How to map and search contents from named-devices corresponding to content requests efficiently?

H. Quality of Service

As ICN-IoT 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-IoT specific applications quality of service (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 an urgent need to design QoS-aware protocols to evaluate the performance of ICN-IoT 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, multicast, adaptability to mobile devices and dynamic environments, and content security at the network layer reduces many efforts that need 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 IoT because IoT 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 IoT.

We identify some main questions that are needed to be explored and answered by the research community from the perspective of major entities involved in the designing of these strategies. From the consumer side, researchers need to investigate the following questions. What benefits will customers receive by sharing the data of their own servers, let us say, data from home server, to be cached? How will the privacy of a consumer be endured? How much a consumer have to pay to upgrade to ICN-based IoT 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 IoT will help to improve the QoS? How it will assist to increase revenue growth? What they would need to offer customers for caching the data? Most importantly, every country government needs to participate in deciding the extent of data sharing.

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

IX. Conclusion

We discussed and presented related literature of both new paradigms IoT and ICN. Then, requirements and challenges to build a reliable and interoperable communication network architecture for IoT are presented. Through this paper, we have also discussed ICN suitable features and different ICN projects for the future Internet design and their resulting ICN-based network architectures for IoT. ICN projects are briefly summarized in terms of their corresponding feasibility for IoT in terms of naming schemes, caching mechanisms, security, and mobility support. Mapping of IoT communication network architecture requirements against ICN striking and supporting features is presented. Furthermore, we discussed ICN-based solutions/architectures for IoT to present the applicability of ICN for IoT. 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, compatible OSs and simulators for ICN-based IoT are discussed in the next section. In the end, we present identified research gaps which need research community attention to build ICN-based network architecture for IoT.

References

[1] *IERC-European Research Cluster on the Internet of Things*. Accessed: Jan. 17, 2016. [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," *Comput. Netw.*, 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 Gener. Comput. Syst.*, 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," Named Data Netw. Project, Rep. NDN-0038, Feb. 2016.
[5] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Comput. Commun.*, vol. 54, no. 1, pp. 1–31, 2014.
[6] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.
[7] V. Varadharajan and S. Bansal, "Data security and privacy in the Internet of Things (IoT) environment," in *Connectivity Frameworks for Smart Devices*. Cham, Switzerland: 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. Ind. Technol. (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 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," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
[11] *Named Data Networking (NDN) Project*. Accessed: Jun. 10, 2015. [Online]. Available: http://named-data.net/
[12] *Pursuing a Pub/Sub Internet-FP7 Project Pursuit*. Accessed: Jun. 12, 2015. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb
[13] *Network of Information (NetInf)*. Accessed: Apr. 17, 2015. [Online]. Available: http://www.netinf.org/
[14] *Comet Project Overview*. Accessed: Apr. 24, 2015. [Online]. Available: http://www.comet-project.org/overview.html
[15] *Fp7convergence Project*. Accessed: Apr. 24, 2015. [Online]. Available: http://www.ict-convergence.eu/
[16] *Mobilityfirst Future Internet Architecture Project*. Accessed: Apr. 24, 2015. [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/
[17] (2016). *Cyber-Secure Data and Control Cloud for Power Grids*. Accessed: Jul. 20, 2016. [Online]. Available: http://cdax.eu/

[18] (2016). *Greenicn Architecture and Applications of Green Information Centric Networking*. Accessed: Jul. 30, 2016. [Online]. Available: http://www.greenicn.org/

[19] *The CCNx Project*. Accessed: Mar. 31, 2015. [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 Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[21] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 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. Commun. (ICC)*, Jun. 2015, pp. 648–653.

[23] N. Fotiou and G. C. Polyzos, "Realizing the Internet of Things using information-centric networking," in *Proc. 10th Int. Conf. Heterogeneous Netw. Qual. Rel. Security Robustness (QShine)*, 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 Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[25] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 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 Things J.*, vol. 3, no. 5, pp. 720–734, Oct. 2015.

[27] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

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

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

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

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

[32] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 122–129, Nov. 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," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 44–53, Dec. 2012.

[34] M. Amadeo *et al.*, "Information-centric networking for the Internet of Things: Challenges and opportunities," *IEEE Netw.*, vol. 30, no. 2, pp. 92–100, Mar./Apr. 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] S. Ziegler, P. Kirstein, L. Ladid, A. Skarmeta, and A. Jara. (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] R. Chen, Y. Wang, Y. Liu, and Z. Chen, "RFID anti-collision algorithm based on tags grouping," *J. Comput. Appl.*, vol. 33, no. 8, pp. 2132–2135, 2013.

[46] M. Collotta and G. Pau, "Bluetooth for Internet of Things: A fuzzy approach to improve power management in smart homes," *Comput. Elect. Eng.*, vol. 44, no. 13, pp. 137–152, 2015.

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

[48] A. Mroué, M. Heddebaut, F. Elbahhar, A. Rivenq, and J.-M. Rouvaen, "Automatic radar target recognition of objects falling on railway tracks," *Meas. Sci. Technol.*, vol. 23, no. 2, p. 10, 2012.

[49] K. Christensen *et al.*, "IEEE 802.3az: The road to energy efficient Ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, Nov. 2010.

[50] *Par for 802.11ah*. Accessed: Jan. 20, 2016. [Online]. Available: http://www.ieee802.org/11/PARs/P802.11ah.pdf

[51] *IEEE Standards Association*. Accessed: Jan. 20, 2016. [Online]. Available: http://standards.ieee.org/news/2014/ieee-802-11ac-ballot.html

[52] T. A. Ramrekha, O. Adigun, A. Ladas, N. Weerasinghe, and C. Politis, "Towards a scalable routing approach for mobile ad-hoc networks," in *Proc. IEEE 20th Int. Workshop Comput.-Aided Model. Design Commun. Links Netw. (CAMAD)*, 2015, pp. 261–266.

[53] R. P. Jover and I. Murynets, "Connection-less communication of IoT devices over LTE mobile networks," in *Proc. 12th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Seattle, WA, USA, 2015, pp. 247–255.

[54] 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 *Proc. IEEE 16th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2015, pp. 1–9.

[55] O. Novo *et al.*, "Capillary networks—Bridging the cellular and IoT worlds," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, 2015, pp. 571–578.

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

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

[58] 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 Things (WF-IoT)*, Dec. 2015, pp. 132–137.

[59] K. E. Skouby and P. Lynggaard, "Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services," in *Proc. Int Contemporary Comput. Informat. (IC3I) Conf.*, Nov. 2014, pp. 874–878.

[60] C. Boldrini, K. Lee, M. Önen, J. Ott, and E. Pagani, "Opportunistic networks," *Comput. Commun.*, vol. 48, pp. 1–4, Jul. 2014, doi: 10.1016/j.comcom.2014.04.007.

[61] 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. Ind. Informat. (INDIN)*, Jul. 2015, pp. 988–992.

[62] 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 Things Cloud (FiCloud) Conf.*, Aug. 2014, pp. 23–30.

[63] *All the Best Big Data Tools and How to Use Them—Import.io*. Accessed: Dec. 30, 2015. [Online]. Available: https://www.import.io/post/all-the-best-big-data-tools-and-how-to-use-them/

[64] K. Kotis and A. Katasonov, "Semantic interoperability on the Internet of Things," *Int. J. Distrib. Syst. Technol.*, vol. 4, no. 3, pp. 47–69, 2013, doi: 10.4018/jdst.2013070104.

[65] (2017). *Integrating Objects to Create New Networked Services*. [Online]. Available: http://www.etsi.org/technologies-clusters/clusters/connecting-things

[66] (2017). *IoT Standards and Protocols*. [Online]. Available: https://www.postscapes.com/internet-of-things-protocols/

[67] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named data networking for IoT: An architectural perspective," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, 2014, pp. 1–5.

[68] (2012). *Information-Centric Networking (ICNRG)*. [Online]. Available: https://irtf.org/rg/icnrg/about/

[69] A. Ghodsi *et al.*, "Information-centric networking," in *Proc. 10th ACM Workshop Hot Topics Netw. (HotNets)*, 2011, Art. no. 1, doi: 10.1145/2070562.2070563.

[70] G. M. Brito, P. B. Velloso, and I. M. Moraes, *Information Centric Networks: A New Paradigm for the Internet*. Hoboken, NJ, USA: Wiley, 2013.

[71] C. Dannewitz, M. D'Ambrosio, and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks," *Comput. Commun.*, vol. 36, no. 7, pp. 736–749, Apr. 2013, doi: 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 *Proc. 11th IEEE/ACM Int. Symp. Model. Anal. Simulat. Comput. Telecommun. Syst. (MASCOTS)*, 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 Commun. Lett.*, vol. 21, no. 4, pp. 873–876, Apr. 2016.

[74] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, Oct. 2007, doi: 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. Conf. Workshop Int Netw. Future (NOF)*, Dec. 2014, pp. 1–5.

[76] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the IoT: Experiments with NDN in the wild," in *Proc. 1st Int. Conf. Inf. Centric Netw.*, 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 Netw.*, vol. 28, no. 3, pp. 50–56, May/Jun. 2014.

[78] S. Li, Y. Zhang, D. Raychaudhuri, and R. Ravindran, "A comparative study of mobilityfirst and NDN based ICN-IoT architectures," in *Proc. Conf. 10th Int. Heterogeneous Netw. Qual. Rel. Security Robustness (QShine)*, 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 Netw. Conf. (IFIP Netw.)*, 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. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, Apr. 2014, pp. 482–487.

[81] J. Quevedo, D. Corujo, and R. Aguiar, "A case for ICN usage in IoT environments," in *Proc. IEEE Glob. Commun. Conf.*, Austin, TX, USA, 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," in *Proc. Wired/Wireless Internet Commun.*, Jan. 2015, pp. 313–326, doi: 10.1007/978-3-319-22572-2_23.

[83] Z. Li *et al.*, "ICN based shared caching in future converged fixed and mobile network," in *Proc. IEEE 16th Int. Conf. High Perform. Switching Routing (HPSR)*, Budapest, Hungary, 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. Conf. Recent Adv. Internet Things (RIoT)*, Singapore, Apr. 2015, pp. 1–6.

[85] S. Vural *et al.*, "In-network caching of Internet-of-Things data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, 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 Comput. Netw. (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. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Turin, Italy, Apr. 2013, pp. 79–80.

[88] *Bonnmotion*. Accessed: Aug. 20, 2016. [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 *et al.*, "Networking named content," in *Proc. ACM 5th Int. Conf. Emerg. Netw. Exp. Technol.*, Rome, Italy, 2009, pp. 1–12.

[92] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[93] *PSIRP Project*. Accessed: Jun. 20, 2015. [Online]. Available: http://www.psirp.org/

[94] *FP7-Sail Project*. Accessed: Jun. 30, 2015. [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 *Proc. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2. Hong Kong, 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, Rep., Apr. 2016.

[97] 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, Rep., Oct. 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 *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Turin, Italy, 2013, pp. 394–398.

[99] J. Burke, P. Gasti, N. Nathan, and G. Tsudik, "Secure sensing over named data networking," in *Proc. IEEE 13th Int. Symp. Netw. Comput. Appl. (NCA)*, Cambridge, MA, USA, 2014, pp. 175–180.

[100] Y. Abidy, B. Saadallahy, A. Lahmadi, and O. Festor, "Named data aggregation in wireless sensor networks," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, May 2014, pp. 1–8.

[101] 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 Veh. Expo (ICCVE)*, Vienna, Austria, Nov. 2014, pp. 765–766.

[102] S. H. Bouk, S. H. Ahmed, and D. Kim, "Hierarchical and hash based naming with compact trie name management scheme for vehicular content centric networks," *Comput. Commun.*, vol. 71, pp. 73–83, Nov. 2015, doi: 10.1016/j.comcom.2015.09.014.

[103] B. Saadallah, A. Lahmadi, and O. Festor, "CCNx for Contiki: Implementation details," Ph.D. dissertation, INRIA, Rocquencourt, France, 2012.

[104] 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

[105] N.-T. Dinh and Y. Kim, "Potential of information-centric wireless sensor and actor networking," in *Proc. IEEE Int. Conf. Comput. Manag. Telecommun. (ComManTel)*, Ho Chi Minh City, Vietnam, 2013, pp. 163–168.

[106] J. Hong, W. Chun, and H. Jung, "A flat name based routing scheme for information-centric networking," in *Proc. IEEE 17th Int. Conf. Adv. Commun. Technol. (ICACT)*, Seoul, South Korea, 2015, pp. 444–447.

[107] B. Li, A. P. Verleker, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for ICN naming scheme," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, San Francisco, CA, USA, 2014, pp. 391–399.

[108] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Social cooperation for information-centric multimedia streaming in highway VANETs," in *Proc. IEEE 15th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Sydney, NSW, Australia, 2014, pp. 1–6.

[109] S. Arshad *et al.*, "Hierarchical and flat based hybrid naming scheme in content-centric networks of things," *IEEE Internet Things*, vol. 5, no. 2, pp. 1070–1080, Apr. 2018.

[110] S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "Comparison of naming schema in ICN," in *Proc. 22nd IEEE Int. Symp. Local Metropolitan Area Netw. (LANMAN)*, Rome, Italy, 2016, pp. 1–6.

[111] Y. Sun, Y. Zhang, H. Zhang, B. Fang, and X. Du, "Geometric routing on flat names for ICN," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

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

[113] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in ICN: Attribute-based encryption and routing," in *Proc. 3rd ACM SIGCOMM Workshop Inf. Centric Netw.*, Hong Kong, 2013, pp. 39–40.

[114] 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 Commun. Lett.*, vol. 18, no. 1, pp. 102–105, Jan. 2014.

[115] A. Compagno, M. Conti, and R. E. Droms, "OnboardiCNg: A secure protocol for on-boarding IoT devices in ICN," in *Proc. ICN*, Kyoto, Japan, 2016, pp. 166–175.

[116] T. Mick, R. Tourani, and S. Misra, "LASeR: Lightweight authentication and secured routing for NDN IoT in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 755–764, 2018.

[117] 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 *Proc. Int. Conf. Future Netw. Distrib. Syst. (ICFNDS)*, Cambridge, U.K., 2017, p. 41. [Online]. Available: http://doi.acm.org/10.1145/3102304.3102345

[118] M. Enguehard, R. E. Droms, and D. Rossi, "SLICT: Secure localized information centric things," in *Proc. ICN*, Kyoto, Japan, 2016, pp. 255–260.

[119] J. Suarez *et al.*, "A secure IoT management architecture based on information-centric networking," *J. Netw. Comput. Appl.*, vol. 63, pp. 190–204, Mar. 2016.

[120] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "A secure ICN-IoT architecture," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Paris, France, 2017, pp. 259–264.

[121] Y. Zhang, D. Raychadhuri, R. Ravindran, and G. Wang, *ICN Based Architecture for IoT*, IETF/IRTF/ICNRG, Oct. 2013.

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

[123] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, and K. Drira, "Producer mobility support in named data Internet of Things network," *Procedia Comput. Sci.*, vol. 109, pp. 1067–1073, Jun. 2017.

[124] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A survey of mobility support in named data networking," in *Proc. 3rd Workshop Name Orient. Mobility Archit. Algorithms Appl. (NOMA)*, San Francisco, CA, USA, 2016, pp. 83–88.

[125] J. Augé *et al.*, "Anchor-less producer mobility in ICN," in *Proc. ACM 2nd Int. Conf. Inf. Centric Netw.*, 2015, pp. 189–190.

[126] J. Augé *et al.*, "MAP-Me: Managing anchor-less producer mobility in content-centric networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 2, pp. 596–610, Jun. 2018.

[127] A. Compagno, X. Zeng, L. Muscariello, G. Carofiglio, and J. Augé, "Secure producer mobility in information-centric network," in *Proc. 4th ACM Conf. Inf. Centric Netw.*, 2017, pp. 163–169.

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

[129] 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 Commun. Lett.*, vol. 20, no. 2, pp. 336–339, Feb. 2016.

[130] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. IEEE 29th Annu. Int. Local Comput. Netw. Conf.*, Tampa, FL, USA, Nov. 2004, pp. 455–462.

[131] *FreeRTOS—Market Leading RTOS (Real Time Operating System) for Embedded Systems With Internet of Things Extensions*. Accessed: Jan. 14, 2016. [Online]. Available: http://www.freertos.org

[132] P. Levis, "Experiences from a decade of TinyOS development," Presented at the 10th USENIX Symp. Oper. Syst. Design Implement. (OSDI), 2012, pp. 207–220.

[133] *Welcome!-OPENWSN—Confluence*. Accessed: Jan. 15, 2016. [Online]. Available: https://openwsn.atlassian.net/wiki

[134] *NS-3 Based Named Data Networking (NDN) Simulator*. Accessed: Jan. 15, 2016. [Online]. Available: http://ndnsim.net/2.1/

[135] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSIM: An highly scalable CCN simulator," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 2309–2314.

[136] L. Saino, I. Psaras, and G. Pavlou, "Icarus: A caching simulator for information centric networking (ICN)," in *Proc. 7th Int. Conf. Simulat. Tools Techn.*, 2014, pp. 66–75, doi: 10.4108/icst.simutools.2014.254630.

[137] M. Tortelli, D. Rossi, G. Boggia, and L. A. Grieco, "CCN simulators: Analysis and cross-comparison," in *Proc. 1st Int. Conf. Inf. Centric Netw.*, Paris, France, 2014, pp. 197–198.

[138] *Get Started With Contiki, Instant Contiki and Cooja*. Accessed: Jan. 28, 2016. [Online]. Available: http://www.contiki-os.org/start.html#start-cooja

[139] 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 Comput. Netw.*, Zürich, Switzerland, Oct. 2009, pp. 834–841.

[140] *The CCNX Project*. Accessed: Mar. 31, 2015. [Online]. Available: http://www.ccnx.org/

[141] *CCNX Packet Format*. Accessed: Mar. 31, 2015. [Online]. Available: https://www.ccnx.org/packet-format/

[142] *Type-Length-Value (TLV) Encoding*. Accessed: Mar. 31, 2015. [Online]. Available: http://named-data.net/doc/ndn-tlv/tlv.html

[143] 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 *Proc. 4th ACM Conf. Inf. Centric Netw.*, Berlin, Germany, 2017, pp. 36–42.

[144] C. Gündogan *et al.*, "Information-centric networking for the industrial IoT," in *Proc. 4th ACM Conf. Inf. Centric Netw.*, Berlin, Germany, 2017, pp. 214–215.

[145] Y. Zhang *et al.*, "ICN based architecture for IoT—Requirements and challenges draft-zhang-iot-icn-challenges-02," ICN Res. Group, Geneva, Switzerland, Internet-Draft, 2016. [Online]. Available: https://tools.ietf.org/html/draft-zhang-iot-icn-challenges-02#page-12

[146] T. Zhang, H. Fan, J. Loo, and D. Liu, "User preference aware caching deployment for device-to-device caching networks," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2017.2773580.

[147] H. Fan, T. Zhang, J. Loo, and D. Liu, "Caching deployment algorithm based on user preference in device-to-device networks," in *Proc. GLOBECOM*, Singapore, 2017, pp. 1–6.

[148] O. Hahm *et al.*, "Low-power Internet of Things with NDN & cooperative caching," in *Proc. ACM ICN 4th ACM Conf. Inf. Centric Netw.*, Berlin, Germany, 2017, pp. 98–108.

[149] X. Jiang, J. Bi, Y. Wang, P. Lin, and Z. Li, "A content provider mobility solution of named data networking," in *Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP)*, Austin, TX, USA, 2012, pp. 1–2.

[150] H. Ko, Y. Kim, D. Suh, and S. Pack, "A proactive content pushing scheme for provider mobility support in information centric networks," in *Proc. IEEE 11th Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2014, pp. 523–524.

[151] D. Saxenaa, V. Raychoudhurya, N. Surib, C. Beckerc, and J. Cao, "Named data networking: A survey," *Comput. Sci. Rev.*, vol. 19, pp. 15–55, Feb. 2016.

[152] A. Lindgren, F. Ben Abdesslem, B. Ahlgren, O. Schelén, and A. M. Malik, "Design choices for the IoT in information-centric networks," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2016, pp. 882–888.

[153] D. Saxena and V. Raychoudhury, "Design and verification of an NDN-based safety-critical application: A case study with smart healthcare," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

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

[155] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.

[156] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin, "An information centric network for computing the distribution of computations," in *Proc. ACM 1st Int. Conf. Inf. Centric Netw.*, Paris, France, 2014, pp. 137–146.

[157] Q. Wang, B. Lee, N. Murray, and Y. Qiao, "CS-MAN: Computation service management for IoT in-network processing," in *Proc. IEEE 27th Irish Signals Syst. Conf. (ISSC)*, 2016, pp. 1–6.

[158] C. Scherb, D. Grewe, M. Wagner, and C. Tschudin, "Resolution strategies for networking the IoT at the edge via named functions," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2018, pp. 1–6.

[159] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang, "Towards software defined ICN based edge-cloud services," in *Proc. IEEE 2nd Int. Conf. Cloud Netw. (CloudNet)*, San Francisco, CA, USA, 2013, pp. 227–235.

[160] E. Borgia, R. Bruno, M. Conti, D. Mascitti, and A. Passarella, "Mobile edge clouds for information-centric IoT services," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Messina, Italy, 2016, pp. 422–428.

Sobia Arshad received the B.Eng. degree in computer engineering and M.S. degree from the University of Engineering and Technology (UET) at Taxila, Taxila, Pakistan, in 2010 and 2013, respectively.

Before joining UET, she was an Assistant Professor with the CS Department, University of Lahore, Lahore, Pakistan, and a Visiting Lecturer with the CS and IT Department, University of Sargodha, Sargodha, Pakistan. She is currently a fulltime Ph.D. Researcher with UET. Her current research interests include information centric networking-based Internet of Things, wireless sensor networks, Internet of Things, smart campus, future Internet, and information centric networking.

Ms. Arshad was a recipient of the full-time scholarship from UET in 2011 and 2014.

Muhammad Awais Azam received the B.Sc. degree in computer engineering from the University of Engineering and Technology (UET) at Taxila, Taxila, Pakistan, the M.Sc. degree (with Distinction) in wireless networks from Queen Mary University, London, U.K., in 2008, and the Ph.D. degree in pervasive and ubiquitous computing from Engineering and Information Sciences Department, Middlesex University, London, U.K., in 2012.

Between 2006 and 2007, he was a Lecturer with UET, where he has been an Assistant Professor with the Department of Computer Engineering since 2013. From 2012 to 2013, he was the Head of Academics with the Cromwell College of IT and Management, London. He leads a research team of M.S. and Ph.D. students in the area of pervasive and ubiquitous computing. His current research interests include network architecture, Internet of Things, network security, embedded systems, ambient intelligence, wireless communications, opportunistic networks, and recommender systems.

Dr. Azam was a recipient of the Gold Medal in 2006.

Mubashir Husain Rehmani (M'14–SM'15) received the B.Eng. degree in computer systems engineering from the Mehran University of Engineering and Technology, Jamshoro, Pakistan, in 2004, the M.S. degree from the University of Paris XI, Paris, France, in 2008, and the Ph.D. degree from Pierre and Marie Curie University, Paris, in 2011.

He is currently with the Telecommunications Software and Systems Group, Waterford Institute of Technology, Waterford, Ireland. He served as an Assistant Professor with the COMSATS Institute of Information Technology, Wah Cantonment, Pakistan, for five years. He has authored/edited two books published by IGI Global, USA, in 2014 and 2015, one book published by CRC Press, USA, in 2016, and one book with Wiley, U.K., in 2018.

Dr. Rehmani was a recipient of the Best Researcher of the Year 2015 of COMSATS Wah Award in 2015, the certificate of appreciation, "Exemplary Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS for the year 2015" from the IEEE Communications Society, and the Best Paper Award from the IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling, the IEEE ICC 2017, and the Higher Education Commission, Government of Pakistan, in 2017, the Research Productivity Award in 2016 and 2017, and has ranked #1 in all Engineering disciplines from the Pakistan Council for Science and Technology, Government of Pakistan. He is currently an Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has served as an Associate Editor for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS from 2015 to 2017. He currently serves as an Associate Editor for the *IEEE Communications Magazine*, *Journal of Network and Computer Applications*, and the *Journal of Communications and Networks*. He is also serving as a Guest Editor for *Ad Hoc Networks Journal*, *Future Generation Computer Systems Journal*, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *Pervasive and Mobile Computing Journal*.

Jonathan Loo received the M.Sc. degree (with Distinction) in electronics and Ph.D. degree in electronics and communications from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and 2003, respectively.

From 2003 to 2010, he was a Lecturer of Multimedia Communications with the School of Engineering and Design, Brunel University, Uxbridge, U.K. From 2010 to 2017, he was an Associate Professor of Communication Networks with the School of Science and Technology, Middlesex University, London, U.K. Since 2017, he has been a Chair Professor of Computing and Communication Engineering with the School of Computing and Engineering, University of West London, London. He has successfully graduated 18 Ph.D. students as their principal supervisor. His current research interests include information centric networking, cloud computing, wireless/mobile networks and protocols, network security, wireless communications, Internet of Things/cyber-physical systems, and embedded systems. He has co-authored over 240 journal and conference papers in the aforementioned specialized areas. He was the Lead Editor of the book entitled *Mobile Ad Hoc Networks: Current Status and Future Trends* (CRC Press, 2011).

Dr. Loo has been an Associate Editor of *International Journal of Communication Systems* since 2011.