ICN Research Group
Internet-Draft
Intended status: Informational
Expires: November 3, 2019

R. Ravindran, Ed.
Huawei Technologies
Y. Zhang
WINLAB, Rutgers University
L. Grieco
Politecnico di Bari (DEI)
A. Lindgren
RISE SICS
J. Burke
UCLA REMAP
B. Ahlgren
RISE SICS
A. Azgin
Huawei Technologies

May 2, 2019

Design Considerations for Applying ICN to IoT
draft-irtf-icnrg-icniot-03

Abstract

The Internet of Things (IoT) promises to connect billions of objects to the Internet. After deploying many stand-alone IoT systems in different domains, the current trend is to develop a common, "thin waist" of protocols to enable a horizontally unified IoT architecture. The objective of such an architecture is to make resource objects securely accessible to applications across organizations and domains. Towards this goal, quite a few proposals have been made to build an application-layer based unified IoT platform on top of today’s host-centric Internet. However, there is a fundamental mismatch between the host-centric nature of today’s Internet and the mostly information-centric nature of the IoT domain. To address this mismatch, the common set of protocols and network services offered by an information-centric networking (ICN) architecture can be leveraged to realize an ICN-based IoT (or ICN-IoT) architecture that can take advantage of the salient features of ICN such as naming, security, mobility, compute and efficient content and service delivery support offered by it.

In this draft, we summarize the general IoT demands, and ICN features that support these requirements, and then discuss the challenges to realize an ICN-based IoT framework. Beyond this, the goal of this draft is not to offer any specific ICN-IoT architectural proposal.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

During the past decade, many Internet of Things (IoT) systems have been developed, and deployed in different domains. The recent trend, however, is to evolve these systems towards a unified IoT architecture, in which a large number of objects hosted by non-interoperable protocol domains can connect to the Internet to enable secure interactions with a diverse set of applications across administrative domain. Note that, here, 'unified' is used to imply a scenario, where all the IoT applications, services, and network functions use a common set of transport APIs and network protocols to interact with each other. Typical IoT applications involve sensing, actuation, processing, and secure content distribution, each of which can occur at different timescales and hierarchical levels that depend on the application requirements. To adapt to different scenarios, IoT systems need to adopt an architecture that can provide (i) pull/ push- and publish/subscribe-based application abstractions, (ii) a common naming framework, (iii) support for payload encryption and signature schemes, and (iv) open APIs as opposed to proprietary APIs that are common in today’s systems. These requirements can pose great challenges for the underlying network and systems. To name a few, the IoT system needs to support 50-100 Billion networked objects [1], many of which are mobile. These objects are expected to have extremely heterogeneous means of connecting to the Internet, often with severe resource constraints. Interactions between the applications and the objects are often real-time and dynamic, requiring strong security and privacy protections. In addition, the IoT system design should offer efficient data exchange schemes that take into consideration the application behavior. For instance, in many IoT applications, data consumers usually need the data sensed from the environment without any reference to the subset of sensors that can provide the requested information.

In short, adopting a general IoT perspective, we first motivate the discussion of ICN for IoT by focusing on well known scenarios. We then discuss the IoT requirements that are generally applicable to many of these well known IoT scenarios. Next we discuss how the current application-layer unified IoT architectures are inefficient to meet the above requirements, and how the key ICN features can make it a better candidate to realize a unified IoT framework. Finally, adopting an ICN perspective, we address the main IoT design challenges and requirements posed towards an ICN-based-IoT system design.

2. Motivating ICN for IoT

ICN offers many features that include name-based networking, content object security, in-network caching, compute, and storage, active mobility support, context-aware networking (see Section 3.6), and support for ad hoc networking. Within the context of an IP based IoT design (IP-IoT), all of these features offered by the ICN have to be realized in an application-specific way demonstrating the compelling nature of ICN to design IoT systems.

To be specific, the features offered by ICN can be used to enable a distributed and intelligent data distribution platform that supports heterogeneous IoT services requiring minimal configuration for device bootstrapping, carrying simpler protocols to aid self-organizing among the IoT elements, and offering natural support for compute and caching logic at strategic points in the network. We outline general advantages of using an ICN-based IoT system design and discuss these from the perspective of the several service scenarios that are difficult to realize over IP today, and whose characteristics arguably match the features offered by ICN.

2.1. Advantages of using ICN for IoT

A key concept of ICN is the ability to name data and services independently from its original location (at which it is stored) and this simplifies caching, and enables decoupling of consumers and producers. Therefore, using ICN to design an architecture for IoT data potentially provides many such advantages compared to using traditional host-centric networks and other new architectures. This section highlights the general benefits that the ICN can provide to an IoT network.

o Naming of Devices, Data and Services: The heterogeneity of the deployed network equipment and offered services by an IoT network leads to a large variety of data, services, and devices. While using a traditional host-centric architecture, only devices or their network interfaces are named at the network level, leaving the task to name data and services to the application layer. This can cause different applications to use different naming schemes, and, as a result, no consistent mapping from application layer names to network names may exist. In many applications common to an IoT network, data and services represent the main objective, and ICN provides an intuitive way to name them in a way that can be utilized at the network layer as well. Communication with a specific device is often secondary, but when needed, the same ICN naming mechanisms can also be used. In such case, network distributes content and provides a service at the same time, instead of only sending data between two named devices. In this context, content and services can be provided by several devices, or a group of devices, hence naming data and services is often more important than naming the devices. This naming mechanism also enables self-configuration of the IoT system.

o Security and privacy: ICN advocates the object security model to secure data in the network. This concept is based on the idea of securing information objects, unlike the session-based security mechanisms, which secure the communication channel between a pair of nodes. ICN provides data integrity through name-data integrity, i.e., the guarantee that the given data corresponds to the name with which it was addressed. Signature-based schemes can additionally provide data authenticity, meaning establishing the origin, or provenance, of the data, for example, by cryptographically linking a data object to the identity of a publisher. Confidentiality can be handled on a per-object basis based on the keys established at the application level. All of this means that the actual transmission of data does not have to be secured, since the same security mechanisms protect the data starting with its generation until its consumption, regardless of its mobility/location (i.e., whether it is in transit over a communication channel or stored in an intermediate cache). In an ICN network, each individual object within a stream of immutable objects can potentially be retrieved from a cache in a different location. However, having a trust relationship with each of these different caches is not realistic. Through name-data integrity, ICN automatically guarantees data integrity to the requesting client regardless of the location, from where it is delivered. The object security model also ensures that the content is readily available in a secure state, and if the device constraints are severe enough that it is not able to perform the required cryptographic operations for object security, then it may be possible to offload this operation to a trusted gateway, to which only a single secure channel needs to be established. ICN can also derive a name from a public key, as the cryptographic hash of a public key also enables it to be self-certifying, in which case, authenticating the resource object does not require an external authority [27][28].

- Distributed Caching and Processing: While caching mechanisms are already used by other types of overlay networks, IoT networks can potentially benefit even more from caching and in-network processing, because of the resource constraints imposed on the devices. Furthermore, wireless bandwidth and power supply can be limited for multiple devices sharing a communication channel, and especially for small mobile devices powered by batteries. In this case, avoiding unnecessary transmissions to retrieve and distribute IoT data from/to multiple places becomes important, hence processing and storing such content in the network can save wireless bandwidth and battery power. Moreover, as for other types of networks, IoT-driven applications requiring shorter delays can also benefit from local caches and services to reduce the delays between content request and delivery.

- Sender/Receiver Decoupling: IoT devices may be mobile and face intermittent network connectivity issues. When a specific data is requested, such data can often be delivered by ICN without any consistent direct connectivity between devices. Apart from using structured caching systems as described previously, information can also be spread by forwarding data opportunistically.

2.2. Service Scenarios

- Smart Mobility: Smart end-user devices and machine-to-machine (M2M) connections are undergoing a significant growth. By 2021, there will be more than 10 billion mobile devices and connections, including smartphones, tablets, wearables, and vehicles [1]. The involved fields for these devices range from medicine and health care to fitness, from clothing to environmental monitoring [42].

In particular, one of the most affected domains is transportation and the so-called Intelligent Transport Systems (ITS) [44]. The objective of ITS is to provide a multi-modal transportation system that embraces public and private municipal, regional, national, and trans-national vehicles and fleets. This extremely heterogeneous ecosystem of transportation means is made available to the users through advanced services that can fulfill the usability requirements, while pursuing system level objectives, and which include: (i) the reduction of CO2 footprint, (ii) the real-time delivery of specific goods, (iii) the reduction of traffic within urban areas, (iv) the provisioning of pleasant journeys to tourists, and (v) the general commitment of satisfactory travel time and experience [121]. Within this context, IoT technologies can play a pivotal role. For instance, they enable advanced design paradigms (e.g., Mobility as a Service (MaaS) [41]) with significant implications on the system architecture [50] or lead to novel approaches to traffic modeling [49]. As a consequence, smart mobility support can be a significant use case scenario for ICN-IoT, where the important ICN features that corroborate mobility support are listed as follows:

* ICN is unique in that it supports both infrastructure- and ad hoc-based communications. This makes it suitable to support communication in vehicular ad hoc networks (VANETS) [19][126], along with supporting communication with the infrastructure components like the road side units to serve the needs of several smart mobility applications. ICN's name based network APIs along with its caching feature enable the system to simultaneously operate over multiple heterogeneous radio interfaces using broadcast, unicast or anycast communication modes.

* ICN offers location independence of content, which allows one to manage consumer mobility in a simpler way than it is with IP. Furthermore, different from Mobile IP, which needs 'triangular routing' to locate moving hosts, ICN envisions a mobile consumer to only re-issue content requests or use network based late-binding functions once the mobile entity handoffs from one attachment point to another [45];

* In ICN, since the content is not bound to a specific location, it can be cached anywhere in the network, thereby adding redundancy to the system. In doing so, if a producer loses connectivity while it is moving, a request for its content can be resolved to an intermediate node en-route to or routed towards a nearby off-path caching node [45];

  * The name based request-response communication paradigm considered for ICN decouples publish/subscribe operations in time and space. Therefore, the involved entities (i.e., publishers and subscribers) do not need to be aware of each other or be connected at the same time [46];

  * The use of an in-network Name Resolution Service (NRS) design allows to identify the current location of or associated with a content name in the network, thanks to its network function, which is responsible for updating the location information of a named entity [58].

From a technological perspective, we can list the open challenges as follows: (i) support for ad hoc communications and interoperability across different IoT technologies, (ii) namespace design that is able to harmonize different ITS standards, (iii) scalable data-sharing model(s) across real-time (and non real-time) traffic sources, (iv) design of travel-centric services based on ICN-IoT, (v) seamless support to mobility, and (vi) content authentication and cryptography.

o Smart Building: Buildings are gaining smart capabilities that allow for enhanced comfort, increased safety and security, and improved energy efficiency [105]. In particular, smart buildings are no longer simple consumer(s) (for energy), but can also be prosumers with on-site energy generation systems. These systems can improve a building's usability towards (i) smart heating, ventilation, and air conditioning (HVAC), (ii) smart lightings, (iii) plug loads, and (iv) smart windows. We can list the main requirements for these sub-systems as follows [105]: (i) context awareness, (ii) support for resource-constrained devices, (iii) interoperability across heterogeneous technologies, and (iv) security and privacy protection. The ICN paradigm can ease the fulfillment of these requirements for one simple reason: smart building services are typically information centric by design. To be specific, any time an autonomic management loop is established within a smart building to control a set of physical variables of interest, the information exchanged between the entities (e.g., users, sensors, actuators, and controllers) do not immediately translate to specific nodes within the building, but can be provided by multiple sensors or gateways. The relevance of ICN in a smart building setting is recognized in the literature as well with reference to the several frameworks deployed in various environments. For instance, in [63], nodes are distributed to different rooms, floors, and buildings of a campus university, and their energy consumption and individual behaviors are monitored. A smart home application is investigated in [107] by evaluating the retrieval delay and packet loss statistics for data.

Moreover, [108] designed and tested lighting control over NDN in a theater setting. In short, within the smart building context, we can list the ICN-specific challenges as follows: (i) design of a scalable namespace for uniquely identifying the information of interest and also host services for actuation, (ii) data-sharing model across heterogeneous systems, (iii) self-organizing functionalities for improving network connections between end-nodes, utilities and the control center, (iv) authentication procedures to grant data confidentiality and integrity.

- Smart Grid: Smart Grid systems are increasingly transforming into cyber-physical systems [18] with the goals of maximum automation towards efficiency and minimal human intervention. The system is a very complex one comprising of power distribution grids, end user applications (e.g. Electric Vehicle (EV) charging systems and appliances), smart monitoring systems (spanning the end users and the power grids), heterogeneous energy producing sources (including prosumers), and load distribution/balancing systems. Current smart grid systems are managed using the centralized Supervisory Control and Data Acquisition (SCADA) frameworks with highly restrictive unidirectional communication support [20]. These systems typically have the following requirements: (i) improved flexibility in distributing energy from the feeder, through real-time reconfiguration of multiple monitoring devices (e.g., phasor measurement units or PMUs) and management operations requiring an efficient data delivery infrastructure; (ii) a large scale data delivery infrastructure capable of supporting latency sensitive applications and inter-connecting heterogeneous end user devices that produce, monitor, and/or consume; (iii) resiliency, which is critical to the operation and protection of the grid; (iv) security, to protect mission critical grid applications from network intrusions; and (v) understanding machine-to-machine traffic patterns for optimal placement of storage and computing to maximize efficiency. Smart grid systems can benefit from ICN in the following ways [21][22]:

  * ICN approach of naming content rather than hosts can ensure that the data generated by one subsystem would be useful for multiple entities. Furthermore, naming content can enable the many-to-many communications model, which is very inefficient in the case of host-centric architectures.

  * ICN features such as in-network computing, storage, and caching enable better use of network resources and can benefit diverse application scenarios that vary from latency tolerant applications with low data rates (e.g., smart grid and energy pricing) to applications observing high data rates with stringent delay/disruption requirements (e.g., synchrophasor

measurements). Also, it is typical for smart grid systems to have applications that consume the same data at different rates, in which case in-network caching and computing can be of significant use.

  * Host-centric networking exposes a mission critical infrastructure like the smart grid infrastructure to intrusion and Denial of Service (DOS) attacks, which are directly related to exposing the IP addresses of critical applications and subsystems. Naming contents, services, or devices, on the other hand, de-couples them from the location, thereby reducing the exposure to being targeted based on a geographical context.

  * ICN's name based networking offers the potential for self-configuration during both the bootstrapping phase and the regular operation of the grid, allowing scalable operation with self-recovery during faults or maintenance tasks in the system.

o Smart Industrial Automation: In a smart and connected industrial environment, equipment with sensors generate large volumes of data during normal operation. This range from the highly time-critical data for real-time control of production processes, to the less time-critical data that is collected by a central cloud environment for control room monitoring, and to pure log data without latency requirements that is mainly kept for a posteriori analysis. Industrial wireless networks are difficult environments with many potential interferences occurring at the same time even as hard reliability and real-time requirements are placed by many applications. This means that the available network capacity is not always high, so it becomes likely for traffic with less stringent delay requirements to experience congestion. One such example is, when errors occur in the production process, a mobile workforce is expected to investigate the problem on-site and they will need high resolution data from the faulty machine(s) as well as other process data from the other parts of the plant. The mobile workforces typically perform their diagnostics or maintenance locally, and they rely on the information acquired from the production system both for safety purposes and to solve any other or related issues in the plant. Furthermore, they rely on both the historical data flow (to pinpoint the root cause of the problems) and the current data flow (to assess the present state of the equipment under control). High resolution measurements are typically generated close to the mobile workforce, while the historic data has to be retrieved from the historian servers. In this scenario, multiple workers involved in the process typically access the same data, possibly with a slight time-shift. The network thus needs to support mobile users to get access to the data flows in a way suitable for their physical location and the task requirements. Introducing ICN functionality into the system can lead to several benefits that enhance the working experience and productivity for the mobile workforce.

* When using ICN, naming of data can be done in a way that corresponds well to the current names often used in industrial scenarios as the hierarchical names defined by the OPC Foundation [10] can be easily mapped to the CCN/NDN name space.

* ICN provides the possibility to get the newest data without knowing the location of the caching nodes or whether a particular piece of data is available locally or in a central repository. ICN also gives the possibility to get either the local high-resolution data or the remote low-resolution data (as there is no need to store all the data centrally, which may not even be possible due to the large data volume). However, it may require well-defined naming conventions or routing policies that can route interests to the right location.

* ICN can reduce the network utilization as unnecessary data is not transmitted, and data accessed by multiple workers is only sent once.

* Workforce mobility between different access points in the factory can be inherently supported, without the need to maintain a connection state.

* Use of ICN can help with removing tedious configurations in clients, since that would be provided by the infrastructure.

* ICN allows the sharing of large volumes of data between users that are in physical proximity, without introducing additional traffic on the backbone network.

* Caching of data in ICN means avoiding additional accesses to a distributed redundant database in the central infrastructure with consistency requirements.

3. IoT Architectural Requirements

Future IoT platforms have to support secure interactions among a large number of heterogeneous, constrained, static or mobile resources across organization/domain boundaries. As a result, it naturally poses stringent requirements in every aspect of the system design. Below, we outline the important requirements that a future IoT platform has to address.

3.1. Naming

An important step towards realizing a unified IoT architecture is the ability to assign names that are unique to (i) each device, (ii) each data item generated by each of these devices, and (iii) each service hosted in a device or a group of devices, towards a common objective. We can assume the naming to have the following requirements. First, names need to be persistent against dynamic features that are common to IoT systems, such as lifetime, mobility, or migration. Second, names that are derived from the keys need to be self-certifying, for both device-centric and content-centric communications. For device-centric communications, binding between device names and the device must be secure. For content-centric communications, binding between the names and the content has to be secure. Third, names usually serve multiple purposes, i.e., routing, security (self-certifying), or human-readability. For IoT applications, the choice of flat versus human readable names needs to be made considering the application and network requirements such as privacy and network level scalability, resource constrained networking requirements, and the name space explosion that may occur because of the complex relationship between name hierarchies [124] that may confound application logic.

One of the challenges in naming is to ensure the trustworthiness of the names. A general approach would require a name certificate service. Such a service acts as a certificate authority in assigning names, which are themselves public keys or appropriately bound to the name for verification at the consumer end.

3.2. Security and Privacy

A variety of security and privacy concerns exist in IoT systems as they are infrastructure typically owned by private entities. For example, the unified IoT architecture makes physical objects accessible to applications across organizations and domains. Furthermore, it often integrates with a critical infrastructure and an industrial system with life safety implications, bringing with it significant security challenges and regulatory requirements [13], as will be discussed in Section 5.3. Security and privacy thus become a serious concern, as does the flexibility and usability of the design approaches. Beyond the overarching trust management challenge, security includes data integrity, authentication, and access control at different layers of the IoT architecture. Privacy includes several aspects: (i) privacy of the data producer/consumer that is directly related to each individual vertical domain such as health, electricity, etc., (ii) privacy of data content, and (iii) privacy of contextual information such as time and location of data transmission [68].

3.3. Scalability

Cisco [1] predicts that there will be around 50 Billion IoT devices on the Internet by 2020 (and these devices include sensors, Radio-Frequency IDentification (RFID) tags, and actuators), and a unified IoT platform needs to name every entity within, which includes these devices, and data and services accessed by/through them. Scalability has to be addressed at multiple levels of the IoT architecture including naming and name resolution, routing and forwarding, and security. Mobility adds further challenges in terms of scalability. Particularly, with respect to name resolution, the system should be able to register/update/resolve a name within a short latency. Additionally, scalability is also affected by the specific IoT system features such as IoT resource object count, state and rate of information update generated by the sensing devices.

3.4. Resource Constraints

IoT devices can be broadly classified as type 1, type 2, and type 3 devices, with type 1 being the most resource-constrained and type 3 being the most resource-rich [47], where the following are considered as the most typical resource types: power, computing, storage, bandwidth, and user interface.

Power constraints of IoT devices limit how much data these devices can communicate, as it has been shown that communications consume more power than other activities for embedded devices [48]. Flexible techniques to collect the relevant information are required, and uploading every single produced data to a central server is not desirable.

Computing constraints limit the type and amount of processing these devices can perform. As a result, more complex processing needs to be done at the cloud servers or at opportunistic points, for instance at the network edge, hence it is important to balance local computation versus communication costs.

Storage constraints of the IoT devices limit the amount of data that can be stored on these devices. This constraint means that unused sensor data may need to be discarded or stored in an aggregated compact form from time to time.

Bandwidth constraints of the IoT devices limit the amount of communication, hence, impose similar restrictions on the system architecture as the power constraints, i.e., one cannot afford to collect every single sensor data generated by the device and/or use complex control plane protocols. It is also worth mentioning that, this constraint also has implications on maintaining idle chatter in the background to maintain connectivity or other volatile service state.

User interface constraints refer to whether the device is itself capable of directly interacting with a user. Possible mechanisms include, via a display and keypad ,LED indicators or requires network connectivity, either locally or globally, to enable human interaction.

The above discussed resources constraints also impact application performance with respect to the end-to-end latency towards sensing or executing control loop based actuation functions.

3.5. Traffic Characteristics

IoT traffic can be broadly classified into local area traffic and wide area traffic. Local area traffic takes place among the nearby devices. For example, neighboring cars may work together to detect potential hazards on the highway, or sensors deployed in a room may collaborate to determine how to adjust the heating level in the room. These local area communications often involve data aggregation and filtering, carry real time constraints, and require fast discovery and association (for the device, data, or service). At the same time, IoT platform has to also support wide area communications. For example, in the case of Intelligent Transportation Systems, realtime video and sensor feeds from the concerned IoT entities can be used towards re-routing operations based on system state, traffic load, availability of freights, weather forecasts, and so on. Wide area communications also require efficient discovery and resolution services for data/services.

While traffic characteristics for different IoT systems are expected to be different, certain IoT systems have been analyzed and shown to have comparable uplink and downlink traffic volumes for some applications such as [2], which means that we have to optimize the bandwidth use and energy consumption in both directions. Furthermore, IoT traffic demonstrates certain periodicity and burstiness [2]. As a result, traffic characteristics of the IoT services have to be properly accounted for during system planning and provisioning.

3.6. Contextual Communication

Many IoT applications rely on dynamic contexts in the IoT system to initiate, maintain, and terminate communication among the IoT devices. Here, we refer to a context as attributes applicable to a group of devices that share some common features, such as their owners may have a certain social relationship or belong to the same administrative group, or the devices may be present near the same proximity. For example, cars traveling on the highway may form a "cluster" based upon their temporal physical proximity to one another as well as the detection of the same event. These temporary groups are referred to as contexts. There are two types of contexts: (i) long-term quasi-static contexts (i.e., contexts based on social contexts as well as stationary physical locations, such as sensors inside a car or a building) and (ii) short-term dynamic contexts (i.e., contexts based on temporary proximity). Between these two classes, short-term contexts are more challenging to support as they require fast formation, update, lookup and association. Therefore, in this draft, our focus will be on the more challenging latter class. In general, IoT applications need to support not only the interactions among the members of a context, but also the interactions across contexts.

3.7. Handling Mobility

There are several degrees of mobility corresponding to different IoT scenarios, ranging from static (as in fixed assets) to highly dynamic (as in vehicle-to-vehicle environments). Furthermore, mobility in an IoT architecture can refer to: (i) data producer mobility, (ii) data consumer mobility, (iii) IoT network mobility (e.g., a body-area network in motion as a person is walking), and/or (iv) disconnection between a source/destination pair (e.g., due to unreliable wireless links). The requirement on mobility support is to deliver IoT data earlier than an application's acceptable delay constraints for all the above considered cases, and if necessary, to negotiate different connectivity or security constraints specific to each mobile context. More detailed discussions on this issue are presented in Section 5.7.

3.8. Storage and Caching

Storage and caching plays a very significant role depending on the type of IoT ecosystem, which is also a function subjected to privacy and security guidelines. Caching is usually needed to increase data availability in the network and for reliability purposes, which is especially useful for wireless access scenarios and with devices experiencing intermittent connectivity to the infrastructure network. Storage is more important for an IoT system, as data is typically stored for long term analysis. Specifically, data is stored at strategic locations in the network to reduce control and computation related overheads. Depending on the application requirements, caching will strictly be driven by application level policies, considering also the privacy requirements. If, for certain type of IoT data, pervasive caching is allowed, then intermediate nodes may not need to always forward a content request to its original creator. Instead, receiving a cached copy would be sufficient for the IoT applications. This approach may greatly reduce the content access latencies.

Considering the hierarchical nature of the IoT systems, ICN architectures can enable a flexible, heterogeneous, and potentially fault-tolerant approach to storage and caching, thereby providing contextual persistence at multiple levels. Within the context of IoT and considering the application requirements, while offering resolution to replicated stored copies, ICN can efficiently support tradeoffs between content security/privacy and regulations.

3.9. Communication Reliability

IoT applications can be broadly categorized into mission critical and non-mission critical applications. For mission critical applications, reliable communication is one of the most important features, as these applications have strong QoS requirements such as low latency and low error rates during information transfer. To support the objective of reliable communications, it is essential for an underlying system to have the following capabilities: (i) seamless mobility support under normal operating conditions, (i) efficient routing in the presence of intermittent connection loss, (iii) QoS aware routing, (iv) support for redundancy at every system level (i.e., device, service, network, storage, etc.), and (v) support for rich and diverse communication patterns, both within an IoT domain (consisting of multiple IoT nodes and one or more gateway nodes to the Internet) and across multiple such domains.

3.10. Self-Organization

Considering the scalability and efficiency requirements, the unified IoT architecture should be able to self-organize to meet various application requirements, e.g., context-driven discovery, which refers to the capability to quickly discover heterogeneous and relevant local/global devices/data/services based on the context. A publish-subscribe service, or a private trust-driven community grouping or clustering scheme, can be used to support this discovery process. For the former case, the publish-subscribe service must be implemented in a way to efficiently support seamless mobility using techniques such as in-network caching and name-based routing. For the latter case, the IoT architecture should be able to discover the private community groups/clusters in a resource efficient way.

Another aspect of self-organization is the decoupling of the sensing infrastructure from the applications. In a typical IoT deployment, various applications run on top of a vast number of IoT devices. It is not an easy task to upgrade the firmware of the IoT devices, and it is also not practical to re-program these IoT devices to accommodate every change in these applications. Therefore, infrastructure and application specific logics need to be decoupled, and a common interface is required (i) to dynamically configure the interactions among the IoT devices and (ii) to easily modify these application logics on top of the sensing/actuating infrastructure [32] [33].

3.11. Ad hoc and Infrastructure Mode

Depending on the presence of a communication infrastructure, an IoT system can operate in an ad-hoc mode or an infrastructure mode, (or use a combination of two). For example, a vehicle may determine to report its location and status information to a server periodically through a cellular connection, or, a group of vehicles may form an ad-hoc network that collectively detects the road conditions around them. In cases, where an infrastructure is sparse, one of the participating nodes may choose to become a temporary gateway node.

The unified IoT architecture needs to design a common protocol that serves both of these modes. Such a protocol should address the challenges that may arise in them: (i) scalability and low latency for the infrastructure mode and (ii) efficient neighbor discovery and ad-hoc communication for the ad-hoc mode. Finally, we note that hybrid modes are very common in realistic IoT systems.

3.12. IoT Platform Management

Service, control and data planes for an IoT platform will be governed by its own management infrastructure, which includes (i) distributed and centralized middleware, (ii) discovery, naming, self-configuring, and analytic functions, and (iii) information dissemination, to achieve the specific IoT system objectives [27][28][29]. Towards this, new IoT management mechanisms and service metrics need to be developed to measure the success of an IoT deployment. Considering an IoT system’s defining characteristics (such as the potential to carry a large number of IoT devices, the objective to save power, mobility, and ad hoc communications), autonomous self-management schemes become very critical. Furthermore, considering its hierarchical information processing deployment model, the platform needs to orchestrate computational tasks based on the involved sensors and the available computation resources, which may change over time. An efficient resource discovery and management protocol is required to facilitate this process. The trade-off between information transmission and processing is another challenge.

4. State of the Art

Over the years, many stand-alone IoT systems have been deployed in various domains. These systems usually adopt a vertical silo architecture and support a small set of pre-designated applications. A recent trend, however, is to move away from this approach, and towards a unified IoT architecture, in which the existing silo IoT systems, as well as the new systems that are rapidly deployed, can coexist. Here, a unified architecture refers to the case, where all the application and network functions use common APIs and network protocols to interact with each other. This will make their data and services accessible to general Internet applications (which is the case for ETSI-M2M [3] and oneM2M [4] standards). In such a unified architecture, resources can be accessed over the Internet and shared across the physical boundaries of an enterprise. However, current approaches to achieve this objective are mostly based on service overlays over the Internet, whose inherent inefficiencies caused by the use of the IP protocol [58] hinders the architecture from satisfying the IoT requirements outlined earlier, particularly in terms of scalability, security, mobility, and self-organization, which are discussed in more details in Section 4.2.

4.1. Silo IoT Architecture

```
                          [IoT Server]
                               |
                               |
                         ______|_______
    _______             {              }
   {       }            {              }
   {IoT Dev}\           {   Internet   }---[IoT Application]
   {_______}  [IoTGW]---{              }
                        {              }
                        {______________}
```

Figure 1:Silo architecture of standalone IoT systems

A typical standalone IoT system is illustrated in Figure 1, which include the devices, applications, gateway and server nodes. Many IoT devices have limited power and computing resources, unable to directly run the normal IP-based access network protocols (i.e., Ethernet, WiFi, 3G/LTE, etc.). Consequently, these devices operate over non-IP protocols to connect to the Internet servers using an IoT gateway. Through the IoT server, applications can subscribe to the data collected by these devices, or interact with them.

There have been quite a few popular protocols for standalone IoT systems, such as DF-1, MelsecNet, Honeywell SDS, BACnet, etc. However, these protocols are operating at a device-level abstraction, rather than an information driven one, leading to a fragmented information and protocol space that requires application level solutions to achieve interoperability.

4.2. Application-Layer Unified IoT Solutions

The current approach to create a unified IoT architecture is to make IoT gateways and servers adopt standard APIs. IoT devices connect to the Internet through standard APIs and IoT applications subscribe/ receive data through standard control/data APIs. Built on top of today’s Internet, this application-layer unified IoT architecture is the most practical approach towards a unified IoT platform. Towards this, there are ongoing standardization efforts including ETSI[3] and oneM2M[4]. IoT service providers can then use such frameworks to build common IOT gateways and servers for their customers. In addition, IETF’s Constrained RESTful Environments (CORE) working group [5] is developing a set of protocols like Constrained Application Protocol (CoAP) [81], that is a lightweight protocol modeled after HTTP [82] and adapted specifically for the IoT. CoAP adopts the Representational State Transfer (REST) architecture with Client-Server interactions. It uses UDP as the underlying transport protocol with reliability and multicast support. Both CoAP and HTTP are considered as the suitable application level protocols for M2M communications, as well as for IoT. For example, oneM2M (which is one of the leading standards for a unified M2M architecture) has protocol bindings to both HTTP and CoAP for its primitives. Figure 2 shows the architecture adopted in this approach.

```
              Publishing----[IoT Server]----Subscribing--
                  |        /    |       \               |
                  |       /     |        \              |
                  |      /______|_______  \             |
 ___________      |     /{              }  publishing   |
{           }     |    | {              }     |         |
{Smart Homes}\    |    | {   Internet   }---------[IoT Application]
{___________}  [IoTGW]---{              }\    |     ________________
                       | {              } \   |    {                }
                       | {______________}  [IoTGW]-{Smart Healthcare}
                       |        |                  {________________}
              Publishing [IoTGW]
                       |    ____|_____
                       |   {          }
                        ---{Smart Grid}
                           {__________}
```

Figure 2: Implementing an open IoT architecture through standardized APIs on the IoT gateways and the server

4.2.1. Weaknesses of the Application-Layer Approach

The above application-layer approach can work with many different protocols, but the system is built upon today's IP network, which has inherent weaknesses towards supporting a unified IoT system. As a result, it cannot satisfy some of the requirements outlined in Section 3, and the reasoning for that is explained as follows:

o Naming: In current application-layer IoT systems, naming scheme is a host-centric one, that is, the name of a given resource/service is linked to the device that can provide it. In turn, device names are coupled to the IP addresses, which are not persistent in mobile scenarios. On the other side, in IoT systems, the same service/resource can be offered by different devices.

o Security and Trust: In IP, security and trust model is based on the session established between two hosts. Session-based protocols rely on the exchange of several messages to establish a secure session. Use of such protocols in constrained IoT devices can have serious consequences in terms of energy efficiency, because transmission and reception of messages are often more costly than the cryptographic operations. This problem may be amplified with the number of nodes that a constrained device has to interact with, due to increase in both the computation cost and the per-session key state managed by the constrained device. Furthermore, because of focusing on securing communication channels rather than managing the data that needs to be secured directly, current trust management schemes can be considered to be relatively weak.

- Mobility: The application-layer approach uses IP addresses as names at the network layer, which hinders the support for device/service mobility or flexible name resolution. Furthermore, the orthogonal Layer 2/3 management, and application-layer addressing and forwarding required to deploy current IoT solutions limit the scalability and management of these systems.

- Resource Constraints: The application-layer approach requires every device to send data to an aggregator, to a gateway or to the IoT server. Resource constraints of the IoT devices, especially in power and bandwidth, can seriously limit the performance of this approach.

- Traffic Characteristics: In this approach, applications are written in a host-centric manner suitable for point-to-point communication. IoT, however, requires multicast support that is challenging for the application-layer based IoT systems today, which have only limited deployment in the current Internet.

- Contextual Communications: The application-layer based IoT approach may not react to dynamic contextual changes in a timely fashion. The main reason is that the context lists are usually kept at the IoT server and they cannot help with efficient routing of requests at the network layer.

- Storage and Caching: The application-layer approach supports application-centric storage and caching but not what ICN envisions at the network layer, or flexible storage that is enabled via name-based routing or lookups.

- Self-Organization: As the application-layer approach is bound to IP semantics, it is considered as topology-based, and, as a result, it cannot sufficiently satisfy the requirement on self-organization. In addition to the topological self-organization, IoT also requires self-organization at the data and service levels [101], which are also not supported by this approach.

- Ad hoc and Infrastructure Mode: As mentioned above, the overlay-based approach lacks self-organization and adaptation to dynamic topology changes, and, therefore, it cannot provide efficient support for the ad hoc mode of communication.

4.2.2. Relation to Delay Tolerant Networking (DTN) architecture and its suitability for IoT

In [23][24], delay-tolerant networking (DTN) has been considered to support future IoT architectures. DTN was initially developed to support information delivery in the presence of network disruptions and disconnections, but it has also been extended to support heterogeneous networks and name-based routing. The DTN Bundle Protocol is able to achieve some of these same advantages and could be beneficially used in an IoT network to, for example, decouple sender and receiver. The DTN architecture is however centered around named endpoints (or endpoint IDs), each of which usually corresponds to a host or a service, and is mainly a way to transport data, while ICN generalizes this notion to named data, hosts and services and offers ways to address IoT application [25] challenges through features such as (information) naming, discovery, request and dissemination. However, endpoint IDs can also be used to identify named content, enabling the use of the bundle protocol as a transport mechanism for an information-centric system. Such a use of the bundle protocol as a transport would still require other components from an ICN architecture such as naming conventions. However, since the exact transport is not a major focus of the issues addressed by this draft, most of the provided discussions are applicable to a generic ICN architecture.

5. ICN Design Considerations for IoT

This section outlines some of the ICN specific design considerations and challenges that must be considered when adopting an ICN design for IoT applications and systems, and describes some of the trade-offs involved to support large scale IoT deployments with diverse application requirements.

Though ICN integrates (i) abstractions at the content, service, and host levels, (ii) name-based routing, and (iii) computation, caching, and storage as part of the network infrastructure, IoT requires special considerations given the heterogeneity of devices and interfaces such as for constrained networking [63][123][125], data processing, and content distribution models to meet specific application requirements, which we identify as challenges in this section.

5.1. Naming Devices, Data, and Services

Even though the ICN approach of named data and services (i.e., device independent naming) is typically desirable when retrieving IoT data, such data-centric naming may also pose certain challenges.

- Naming of devices: Naming devices [127] [128]can be useful in an IoT system. For example, actuators may require clients to act on a specific node of the deployed network (to switch it on or off), or it could be necessary to access a particular device for administration purposes. This can only be achieved through a specific name that uniquely identifies the targeted network entity. Moreover, a persistent name allows a device to change its attachment point without loosing its identity. A friendly way to address a device is to use a contextual hierarchical name, which is of the same type as one that is used for data objects. Also note that, through disabling of caching and request aggregation on names associated with a device, it is possible to ensure that the requests targeting that device always reach the device.

- Size of data/service name: Content names can have variable lengths. Since each name has to uniquely identify the content and can also include self-certifying properties (e.g., the hash of the content is bound to the name), their lengths can be quite long in relation to the size of the content itself. In particular, for specific application, content name size can even exceed the Data size. This can be the case for IoT networks with sensed values that usually consist only of a few bytes (i.e., data can be as small as a short integer in case of temperature values, or one-byte in case of control messages corresponding to an actuator state as on/off). Moreover, a name that is too long is likely to trigger fragmentation at the link layer, and create additional problems (i.e., several transmissions, increased delay and security issues). Various approaches have been investigated to handle fragmentation and reassembly issues associated with ICN packets. For instance, the work in [109] proposes to perform hop-by-hop operations, i.e., each hop fragments the packet that has to be forwarded and reassembles the packet received for further processing. This mechanism allows to efficiently handle the recovery of lost or corrupted fragments locally, thereby reducing packet delivery failures that require application-level retransmissions.

- Hash-based content name: Hash algorithms are commonly used to name content, in order to verify that the received content is the one requested. This is only possible in contexts, where the requested object already exists, and where there is a directory service to look up names or the names are learned through a manifest service. This approach is suitable for systems with large sized data objects, where it is important to verify the content.

- Hierarchical names: The use of hierarchical names, as is the case with the CCN and NDN architectures, makes it easier to create names a priori based on a predefined naming convention. It also provides a convenient way to use the same naming scheme for device names. However, since names are not self-certifying, this will require other mechanisms for verification of object integrity. If routing is also performed on the hierarchical names, the system will lose some of its location independence and caching will mostly be done on the path towards the publisher.

- Semantic and metadata-based content name: A semantic-based naming approach can allow for successful retrieval of name through a set of keywords (for example, 'noise level at position X'), even if a perfect matching of the name is not available [65]. Moreover, enriching contents with metadata allows to better describe the names and to establish association between similar ones. However, this mechanism requires more advanced functionality to match such metadata in the data object to the semantics of the name (e.g., comparing the position information of an object with the position information of the requested name). The need for such (potentially) computationally heavy tasks at the intermediate nodes in the network may be considered to understand the trade-offs between application and network performance. [64] proposes a metadata-based naming approach to support ICN-IoT networking with service function identification and processing of IoT data at some vantage points in the local IoT network, before returning the processed result to the consumers.

- Naming of services: Similar to naming of devices or data, services can also be referred to with a unique identifier, provided by a specific device or by an authorized entity (i.e., someone assigned by a central authority as the service provider). It can also be a service provided by anyone meeting certain metadata conditions. Example of services may include content retrieval, which takes a content name or description as an input and returns the value of that content, and actuation, which takes an actuation command as an input and possibly returns a status code.

- Trust: Names can be used to verify the authenticity and the integrity of the data. Multiple approaches can be used to provide security functionalities through names. For instance, hierarchical, schematized, Web-of-Trust models can enable public key verification, whereas self-certifying names can enable in-network integrity checks of the name-key or name-content binding without the need of a Public Key Infrastructure (PKI) or another third party to establish whether the key is trustworthy or not. This can be realized either directly or indirectly. In the former case, the hash of the content is bound to the name. In the latter case, first, the hash of the content is signed with the secret key of the publisher, and then the public key of the publisher and the signed hash are bound to the name [46]. The hash algorithm can be applied to the already existing contents and where there is a directory service or manifest to look up names. In case of yet-to-be-published but on-demand generated contents, the hash cannot be known a-priori, hence different trust mechanisms should be investigated. Furthermore, self-certified naming approach can hide the content semantics, thus making names less human friendly. Since trends show that users prefer to find contents through a search engine using keywords, having non-human-friendly names can be a barrier, unless the content is enriched with keywords. However, this problem does not concern M2M applications, as human-readable names may not be useful in the context of just communicating machines.

- Flexibility: Further challenges may arise for the hierarchical naming schema, associated with the requirements on "constructible names" and "on-demand publishing" [37][38]. The former entails that each user is able to construct the name of a desired data item through specific algorithms and that it is possible to retrieve information using partially specified names. The latter refers to the possibility of requesting a not-yet-published content, thereby triggering its creation.

- Scoping: From an application's point of view, scopes are used to gather related data, whereas from the network's perspective, scopes are used to mark where the content is available [68]. For instance, nodes that are involved with caching coordination can vary according to scope [69]. As a result, scoping can be used (i) to limit propagation of requests, thereby improving resource usage efficiency by reducing bandwidth and energy consumption, and (ii) to control content dissemination thanks to access control rules, which can be different for each scope [67]. Note that, relying on scoping for security/privacy has been shown to not work all that well for IP, and is unlikely to work well for ICN either. However, scoping may be useful in certain scenarios, for instance, to limit propagation of requests and provide a simple means to attain context-sensitive communications. Finally, perimeter- and channel-based access control is often violated by the current networks to enable over-the-wire updates and cloud-based services, so scoping is unlikely to replace a need for data-centric security in ICN.

- Confidentiality: As names can reveal information about the nature of the communication (which may also violate the privacy requirements), mechanisms for name confidentiality should be available in the ICN-IoT architecture. To grant confidentiality protection, some approaches have been proposed in order to handle access control in an ICN naming scheme such as Attribute-Based Encryption [66] and access control delegation [67]. In the first solution, a trusted third party assigns a set of attributes to each network entity. Then, a publisher performs the following operations in order: (i) encrypting the data with a random key, (ii) generating the metadata for the decryption phase, (iii) creating an access policy that is used to encrypt the random key, and (iv) appending the encrypted key to the content name. When the consumer receives the packet, if its attributes satisfy the hidden policy in the name, it can get the random key protected in the name and decrypt the data. The second solution introduces a new trusted network entity (i.e., Access Control Provide). In this case, when a publisher generates a content, it also creates an access control policy and send it to an Access Control Provider. This network entity stores the access control policy, to which it associates a Uniform Resource Identifier (URI). This URI is sent to the publisher and included in the advertisements of the content. Then, when a subscriber tries to access a protected content, it can authenticate himself and request authorization for the particular policy to the Access Control Provider through the URI.

5.2. Name Resolution

Inter-connecting numerous IoT entities, as well as establishing reachability to them, requires a scalable name resolution system considering several dynamic factors like mobility of end points, service replication, in-network caching, failure or migration [59] [72] [73] [95]. The objective is to achieve scalable name resolution handling static and dynamic ICN entities with low complexity and control overhead. In particular, the main requirements/challenges of a name space (and the corresponding Name Resolution System where necessary) are [52] [54]:

- Scalability: The first challenge faced by ICN-IoT name resolution system is its scalability. Firstly, the approach has to support billions of objects and devices that are connected to the Internet, many of which are crossing administrative domain boundaries. Second of all, in addition to objects/devices, the name resolution system is also responsible for mapping IoT services to their network addresses. Many of these services are based upon contexts, hence dynamically changing, as pointed out in [59]. As a result, the name resolution should be able to scale gracefully to cover a large number of names/services with wide variations (e.g., hierarchical names, flat names, names with limited scope, etc.). Notice that, if hierarchical names are used, scalability can be also supported by leveraging the inherent aggregation capabilities of the hierarchy. Advanced techniques such as hyperbolic routing [89] may offer further scalability and efficiency.

- Deployability and inter-operability: Graceful deployability and interoperability with existing platforms is a must to ensure a naming schema to gain success on the market [7]. As a matter of fact, besides the need to ensure coexistence between IP-centric and ICN-IoT systems, it is required to make different ICN-IoT realms, each one based on a different ICN architecture, to inter-operate.

- Latency: For real-time or delay sensitive M2M application, the name resolution should not affect the overall QoS. With reference to this issue it becomes important to circumvent too centralized resolution schema (whatever the naming style, i.e, hierarchical or flat) by enforcing in-network cooperation among the different entities of the ICN-IoT system, when possible [99]. In addition, fast name lookup are necessary to ensure soft/hard real time services [110][111][112]. This challenge is especially important for applications with stringent latency requirements, such as health monitoring, emergency handling and smart transportation [113].

- Locality and network efficiency: During name resolution the named entities closer to the consumer should be easily accessible (subject to the application requirements). This requirement is true in general because, whatever the network, if the edges are able to satisfy the requests of their consumers, the load of the core and content seek time decrease, and the overall system scalability is improved. This facet gains further relevance in those domains where an actuation on the environment has to be executed, based on the feedbacks of the ICN-IoT system, such as in robotics applications, smart grids, and industrial plants [101].

- Agility: Some data items could disappear while some other ones are created so that the name resolution system should be able to effectively take care of these dynamic conditions. In particular, this challenge applies to very dynamic scenarios (e.g., VANETs) in which data items can be tightly coupled to nodes that can appear and disappear very frequently.

5.3. Security and Privacy

Security and privacy is crucial to all the IoT applications including the use cases discussed in Section 2 and subjected to the information context. To exemplify this, in one recent demonstration, it was shown that passive tire pressure sensors in cars could be hacked adversely affecting the automotive system [77], while at the same time this and other car information can be used by a public traffic management system to improve road safety. The ICN paradigm is information-centric as opposed to state-of-the-art host-centric

Internet. Besides aspects like naming, content retrieval and caching this also has security implications. ICN advocates the model of trust in content rather than a direct trust in network host mode. This brings in the concept of Object Security which is contrary to session-based security mechanisms such as Transport Layer Security (TLS)/Datagram Transport Layer Security (DTLS) prevalent in the current host-centric Internet. Object Security is based on the idea of securing information objects unlike session-based security mechanisms which secure the communication channel between a pair of nodes for unicast, (or among a set of nodes for multicast/broadcast). This reinforces an inherent characteristic of ICN networks i.e. to decouple senders and receivers. Even session based trust association can be realized in ICN [86], that offers host-independence allowing authentication and authorization to be separated from session encryption, allowing multiple end points to meet specific service objectives. In the context of IoT, the Object Security model has several concrete advantages. Many IoT applications have as its main objective generating data and providing some services, while the communication between two devices is a secondary task. Therefore, it makes more sense to secure IoT objects instead of securing the session between communicating endpoints. Though ICN includes data-centric security features the mechanisms have to be generic enough to satisfy multiplicity of policy requirements for different applications. Furthermore security and privacy concerns have to be dealt in a scenario-specific manner with respect to network function perspective spanning naming, name-resolution, routing, caching, and ICN-APIs. The work by the JOSE WG [83] provides solution approaches to address some of these concerns for object security for constrained devices and should be considered to see what can be applied to an ICN architecture. In general, we feel that security and privacy protection in IoT systems should mainly focus on the following aspects: confidentiality, integrity, authentication and non-repudiation, and availability. Even though, implementing security and privacy methods in IOT systems faces different challenges than in other systems, like IP. Specifically, below we discuss the challenges in the constrained and infrastructure part of the network.

o In resource-constrained nodes, energy limitation is the biggest challenge. Moreover, a node it has to deliver its data over a wireless link for a reasonable period of time on a coin cell battery. As a result, traditional security/privacy measures are impractical to be implemented in the constrained part. In this case, one possible solution might be utilizing the physical wireless signals as security measures [78] [57].

o In the infrastructure part, we have several new threats introduced by ICN-IoT [88] particularly in architectures employing name resolution service [123]. Below we list several possible attacks to a name resolution service that is critical to ICN-IoT:

1. Each IoT device is given an ICN name. The name spoofing attack is a masquerading threat, where a malicious user A claims another user B's name and attempts to associate it with A's own network address NA-A, by announcing the mapping (ID-B, NA-A). The consequence of this attack is a denial of service as it can cause traffic directed for B to be directed to A's network address.

2. The stale mapping attack is a message manipulation attack involving a malicious name resolution server. In this attack, if a device moves and issues an update, the malicious name resolution server can purposely ignore the update and claim it still has the most recent mapping. Perhaps worse, a name resolution server can selectively choose which (possibly stale) mapping to give out during queries. The result is a denial of service.

3. The third potential attack, false announcement attack, is an information modification attack that results in illegitimate resource consumption. User A, which is in network NA1, claims its ID-A binds to a different network address, (ID-A, NA2). Thus A can direct its traffic to network NA2, which causes NA2's network resources to be consumed.

4. The collusion attack is an example of an information modification attack in which a malicious user, its network and the location where the mapping is stored collude with each other. The objective behind the malicious collusion is to allow for a fake mapping involving a false network address to pass the verification and become be stored in the storage place.

5. An intruder may insert fake/false sensor data into the network. The consequence might be an increase in delay and performance degradation for network services and applications.

- IoT data is collected and stored on such servers, which usually run learning algorithms to extract patterns from such data. In this case, it is important to adopt a framework that enables privacy-preserving learning techniques. The framework defines how data is collected, modified (to satisfy the privacy requirement), and transmitted to application developers.

5.4. Caching

In-network caching helps bring data closer to the consumers, but its usage differs in constrained and infrastructure parts of the IoT network. Furthermore, caching in ICN-IoT faces several challenges:

- Which nodes on the routing path should cache the data: According to [54], caching the data on a subset of nodes can achieve a better gain than caching it on every en-route routers. In particular, the authors propose a "selective caching" scheme to locate those routers with better hit probabilities to cache data. According to [55], selecting a random router to cache data is as good as caching the content everywhere. In [91], the authors suggest that edge caching provides most of the benefits of in-network caching but with simpler deployment. However, the existing research on this topic typically consider workloads that are analogous to today's CDNs, rather than the workload that can be attributed to IoT applications considered here. Therefore, further work is needed to understand the appropriate caching approach for IoT applications.

- What to cache for the IoT applications: In many IoT applications, customers often access a stream of sensor data, and as a result, caching a particular sensor data for longer periods may not be beneficial. In [93], a caching scheme is proposed to ensure that older instances of the same sensor stream were first to be evicted from the cache when needed. In [57], the authors suggest to cache IoT services at the intermediate routers, and in [59], the authors suggest to cache the control information such as pub/sub lists at the intermediate nodes. In addition, it is not yet clear what caching means in the context of actuation in an IoT system. For example, it could mean caching the result of a previous actuation request (using other ICN mechanisms to suppress the repeated actuation requests within a given time period), or it could have little meaning at all if the actuation uses authenticated requests as in [92].

- Efficiency of distributed caching may be application dependent: When content popularity is heterogeneous, some content is often requested repeatedly. In this case, the network can definitely benefit from caching. Another case where caching would be beneficial is when devices with low duty cycle are present in the network and when the access to the cloud infrastructure is limited. In [93], it is also shown that there are benefits to caching in the network when edge links are lossy, in particular if the losses occur close to the content producer, as is common for the wireless IoT networks. Furthermore, IoT devices can collaborate to cache content in a manner that optimizes energy efficiency and content availability [94]. However, using distributed caching mechanisms in the network is not useful when each object is only requested at most once, as a cache hit can only occur for the second and subsequent requests. It may also be less beneficial to have caches distributed throughout the ICN network, especially in cases when there are overlays of distributed repositories, e.g., a cloud or a Content Distribution Network (CDN), from which all clients can retrieve the data. Using ICN to retrieve data from such services may add some efficiency, but in case of dense occurrence of overlay CDN servers the additional benefit of caching in ICN nodes would be lower. Another example is when the name refers to an object with dynamic content/state. For example, when the last value for a sensor reading is requested or desired, the returned data may change any time the sensor reading is updated. In such case, in-network caching may increase the risk of returning old or stale data.

5.5. Storage

Storage is useful for IoT systems regardless of its type, be it as a long-term storage or as a short-term storage.

In the case of long-term in-network storage, resources can be distributed among vantage points, which include the network edge and the main IoT service aggregation points such as in the data centers. The main differences, in regards to IoT-driven storage, between the two locations are the size of data, processing intelligence and heterogeneity of information that has to be dealt at these locations. Specifically, the purpose of long term storage at the edge is to analyze, filter, aggregate, and re-publish IoT data for consumption either by the parent service components or directly by the consumers. At the aggregation service points, the purpose is to re-publish the data that will be presented as part of the global pub/sub service to the interested consumers. Long term storage for IoT data also serves the purpose of backup and replication of data, which come with additional caveats. First, we need to decide on the number of replicas needed for each IoT data stream, and the storage locations for these replicas. Also note that, given that many IoT applications consume data locally, storage locations should be kept near the data sources. However, since IoT data is mostly appended to the end of a stream, instead of being updated, it becomes easier to manage these multiple replicas. Second, we need to adopt a mechanism that can efficiently route traffic to the nearest data replica. ICN provides several solutions to this problem, e.g., global name resolution service (GNRS), which can keep track of each replica's location [58].

In the case of short-term in-network storage (where the term storage refers to a temporary buffer, when an outgoing link is not available), the objective is to improve communication reliability, especially when network links are unreliable, such as wireless links. ICN-IoT can adopt a generalized storage-aware routing algorithm to support delay and disruption tolerant packet forwarding. In such case, each router can employ the in-network storage to facilitate store vs. forward decisions in response to varying link conditions and potential network interruptions [115]. These decisions can be based on both short-term and long-term path quality metrics. Additionally, packets along disconnected paths can be handled using a disruption tolerant networking (DTN) based approach to offer delayed delivery and replication features. In particular, each router maintains two types of topology information: (i) an intra-partition graph that is formed by collecting flooded link state advertisements, which carry fine-grained, time-sensitive information about the intra-network links, and (ii) a DTN graph that is maintained via epidemically disseminated link-state advertisements, which carry connection probabilities among all the network nodes. However, for this scenario, we observe the following challenges: (i) when and how long to store the data, and (ii) the next step after the short-term storage. In [93] the authors show that it is beneficial to store data even for shorter periods of time (and even if only a single requester exist), if the network is lossy such that retransmissions and error recovery can be done locally instead of end-to-end.

5.6. Routing and Forwarding

ICN-IoT supports both device-to-device (D2D) and device-to-infrastructure (D2I) communications. D2D communications may occur within a single IoT domain, or across IoT domains, and may involve data forwarding within the source IoT domain, in the infrastructure network, and within the destination IoT domain. D2I communications involve data forwarding within the source IoT domain and in the infrastructure network. Data forwarding within an IoT domain can adopt routing protocols such as RPL [84], AODV[85], etc, with the main challenge being the resource constraints of the IoT nodes. In order to address this challenge, we can adopt a light-weight protocol using much shorter ICN names for each communicating party within an IoT domain (see Section 5.12 for details). In such case, before a packet leaves the IoT domain, gateway node translates this short ICN name associated with the source device to its original ICN name.

At the ICN infrastructure, data forwarding can adopt one of two approaches: (i) direct name-based routing or (ii) indirect name resolution service (NRS) driven routing.

o In direct name-based routing, packets are forwarded using the name corresponding to either the data itself [95][63][74] or the name of the destination node [75]. Here, the main challenge is to keep the state information required for data routing/forwarding at the ICN router small. This can become an especially challenging issue, if the architecture uses a flat naming scheme due to lack of aggregation capabilities.

- In indirect routing, packets are forwarded based on the locator of the destination node, which is obtained through a name resolution service. Here, name-locator binding can be done either before routing (i.e., assuming static binding) or during routing (i.e., assuming dynamic binding). In the case of static binding, router state is the same as that in traditional routers, and the main challenge is to perform name resolution fast, especially with mobile IoT devices. In the case of dynamic binding, ICN routers need to maintain a name-based routing table, and the challenge becomes keeping the state information small, while at the same time performing fast name resolution.

5.7. Mobility Management

Considering the diversity of IoT applications, mobility scenarios range from tracking sensor data from mobile human beings to large fleets of diverse mobile elements such as drones, vehicles, trucks, trains (each of which may be associated with a transport infrastructure). These mobility scenarios can take place over heterogeneous access infrastructure that ranges from short range 802.15.4 communications to cellular radios. It is therefore expected that handling information delivery in an ad hoc setting, which involves vehicles, road side units (RSU), and the corresponding infrastructure based services, shall offer more challenges. ICN architectures have been generally shown to handle consumer and producer mobility scenarios efficiently [61][129], and to be suitable for V2V scenarios [62]. Networking tools to handle mobility varies based on application requirements, which vary from delay and loss tolerant to mission critical (with stringent delay and loss requirements).

Therefore, the challenge becomes to quantify the cost associated with mobility management in both the control and the forwarding planes, to handle both static binding and dynamic binding (which enables seamless mobility) of a named resource to its location when either or both of consumer and/or producer is mobile.

During a network transaction, either the producer or the consumer may move away, and thus we need mechanisms that can handle the mobility of either or both to avoid information loss. ICN differentiates the mobility of a consumer (Case I) from that of a producer (Case II):

- Case I: When a consumer moves to a new location after sending out a request for data, the data may traverse the path towards the previous point of attachment (PoA), and in doing so, leaving copies of it along that path. The data can then be retrieved by the consumer by simply reissuing its request for the data, which is a technique used by the direct routing approach. Conversely, indirect routing approach does not differentiate between consumer and producer mobility [95], as the indirect routing approach only requires an update on the NRS, which can then update the routers to re-bind the named resource to its new location, while using late-binding to route the packet from the previous PoA to the new one.

- Case II: In the case of a producer that has moved, the challenge becomes managing the control overhead while searching for a new data producer (or for re-locating the initial producer) [60]. For this purpose, flooding techniques can be used to re-discover the producer, or direct routing techniques can be employed after enhancing them with the late-binding feature that enables seamless mobility [61].

5.8. Contextual Communication

ICN enables contextualized communications by allowing metadata to be included within control or application payload. Doing so can help IoT applications to adapt to different environments, thereby enabling intelligent networks that are self-configurable and intelligent networking among consumers and producers [57]. For example, let us look at the following smart transportation scenario: "James walks on an NYC street and wants to find an empty taxi closest to his location." In this example, the context is the location information corresponding to James and the taxi drivers. A context service, as an IoT middleware, processes the contextual information and bridges the gap between raw sensor information and application requirements. Alternatively, we can use naming conventions that allow applications to request content in namespaces related to their local context without requiring a specific service, such as /local/geo/ mgrs/4QFJ/123/678 to retrieve objects published within a 100m grid area of 4QFJ 123 678 based on the military grid reference system (MGRS). In both cases, trust providers may emerge that can vouch for an application’s local knowledge.

However, extracting contextual information on a real-time basis can become very challenging:

- First, we need to have a fast context resolution service, through which the subscribed IoT devices can continuously update their contextual information to the application (e.g., for the example above, that would be the locations of James and the taxis). Or, in the case of a namespace driven approach, we need to have mechanisms that can query the nearest neighbor based on a given namespace on a continual basis.

- The difficulty of this challenge grows rapidly as the number of involved devices as well as the number of contexts increase.

5.9. In-network Computing

In-network computing enables ICN routers to host heterogeneous services catering to various network functions and applications needs. Contextual services for IoT networks require in-network computing, with each sensor node or ICN router implementing context reasoning [57]. Another major target for in-network computing is to filter (and cleanse) the sensed data for IoT applications, as the sensed data can be noisy [76].

Within this framework, Named Function Networking (NFN) [117] is proposed as an extension of the ICN concept to named functions, which are processed in the network, and which can be used to generate data flow processing applications (for instance, one that is well-suited to time series data processing by IoT sensing applications). Related to this is the need to support efficient function naming, with functions, input parameters, and the output result can all be encapsulated within the packet header, the packet body, or a mixture of the two (e.g. [33]). If functions are encapsulated within the packet header, naming scheme can impact (i) how a computation task is routed within the network, (ii) which IoT devices are involved with the computation task (e.g. [56]), and (iii) how a name is decomposed into smaller computation tasks and deployed in the network to achieve better performance.

Another challenge is related to how to support compute-aware routing. Default routing is typically used for forwarding requests towards the nearest cache (or source/repository) and return the matching data to the requester. Compute-aware routing, on the other hand, has a different purpose. For instance, if the computation task is for aggregating the sensed data, then the routing strategy becomes routing the data to achieve a better aggregation performance [53].

In-network computing also includes synchronization challenges. Some computation tasks, for instance, may need synchronization among sub-tasks or IoT devices. For instance, a device may not send the generated data as soon as it is available, because waiting for data from the neighbouring devices can lead to better aggregation performance. Or, some devices may choose to sleep to save energy, while waiting for the results from the neighbours. Furthermore, while aggregating the computation results along the path, intermediate IoT devices may need to choose the results generated within a certain time window.

5.10. Self-Organization

General IoT deployments involve heterogeneous IoT systems that consist of embedded systems, aggregators and service gateways in an IoT domain. To scale the IoT deployments to a large scale, scope-based self-organization is typically required. This specifically relates to the IoT system middleware functions [122] that include (i) device bootstrapping and discovery, (ii) assigning local/global names to device and/or content, and (iii) security and trust management functions towards device authentication and data privacy. ICN based on-boarding protocols have been studied [100] and has been shown to offer significant savings compared to the existing approaches. These challenges span both the constrained devices as well as the interactions among the aggregators and the service gateways, which may need to contact external services like the authentication servers to on-board these devices. A critical performance optimization metric for these functions, while operating at scale, is to have low control/data overhead in order to maximize the energy efficiency. Furthermore, within the infrastructure part of the network, scalable name-based resolution mechanisms, pub/sub services, storage and caching, and in-network computing techniques should be studied to meet the scope-based content dissemination needs of an ICN-IoT system.

5.11. Communications Reliability

ICN offers many ingredients for reliable communication, such as multi-home interest anycast over heterogeneous interfaces, caching, and forwarding intelligence for multi-path routing that leverage state-based forwarding in protocols like CCN/NDN. However, these features have not been analyzed from the QoS perspective, when heterogeneous traffic patterns are multiplexed at a router. In general, QoS for ICN is an open area of research [125]. In-network reliability comes at the cost of a complex network layer, hence a research challenge here is to build redundancy and reliability at the network layer to handle a wide range of disruption scenarios, such as congestion, short/long term connection loss, or wireless impairments along the last mile. An ICN network should allow features such as opportunistic store-and-forward mechanisms to be enabled only at certain points in the network, as these mechanisms entail additional control/forwarding plane overheads that can adversely affect the application throughput. For additional details, see Section 5.5, for the discussion on in-network storage.

5.12. Resource Constraints and Heterogeneity

An IoT architecture should take into consideration the resource constraints of the often embedded IoT nodes. Having globally unique IDs (GUID in short) is a key feature in ICN, and these IDs may consist of tens of bytes. Each device would then have a persistent and unique ID no matter when and where it moves. It is also important for ICN-IoT to keep this feature. However, always carrying the long ID in the packet header may not be always feasible, for instance, for transmissions over a low-rate layer-2 protocol such as 802.15.4. To solve this issue, ICN can operate using a lighter-weight packet header and a much shorter locally unique ID (LUID in short). In this way, we can map a device's long GUID to its short LUID when the packet targeting the device reaches the local area IoT domain. To cope with collisions that may occur with this mapping process, we let each domain to have its own GUID-to-LUID mapping scheme, which can be managed by a gateway deployed at the edge of the domain. Different from NAT and other existing domain- or gateway-based solutions, ICN-IoT does not change the identity of an application. The applications, either on the constrained IoT devices or on the infrastructure nodes, continue to use the long GUIDs to identify one another, while the network performs the translation, which is transparent to these applications. An IoT node carries its GUID no matter where it moves, even when it is relocated to another local IoT domain and assigned a new LUID. This ensures the global reachability under mobility, while taking into consideration the resource constraints of the embedded devices.

In addition, optimizations for the other components of the ICN-IoT system (described in earlier subsections) can lead to optimization of the energy consumption as well.

6. Differences from T2TRG

Thing-to-Thing Research Group (T2TRG) [9] is an IoT research group under IRTF, which focuses on the research challenges of realizing IoT solutions assuming IP as the narrow waist. As IP-IoT has been a research topic for over a decade and with active industry solutions, this group provides a venue to study the advanced issues related to its security, provisioning, configuration and inter-operability considering the various heterogeneous application environments. ICN-IoT, on the other hand, is a recent research effort, where the objective is to exploit the ICN features of name based routing and security, caching, multicasting, mobility, etc. in an end-to-end manner to enable IoT services spanning all kind of networking scenarios, i.e., ad hoc, infrastructure, and hybrid scenarios. More detailed comparison of IP-IoT versus ICN-IoT is presented in Section 4.

7. Security Considerations

ICN puts security in the forefront of its design, which the ICN-IoT designs can leverage to build applications with varying security requirements. This issue has been discussed quite elaborately in this draft. However, as this is an informational draft and it does not create new considerations beyond what has been discussed.

8. Conclusions

This draft offers a comprehensive view of the benefits and design challenges of using ICN to deliver IoT services, not only because of its suitability for constraint networks but also for ad hoc and infrastructure environments. The draft begins by motivating the need for ICN-IoT by considering popular IoT scenarios and then delves into understanding the IoT requirements from both application and networking perspectives. We then discuss why the current IP-based application layer unified IoT solutions fall short of meeting these requirements, and how an ICN architecture is more suitable towards addressing the IoT service needs. We then elaborate on the design challenges in realizing an ICN-IoT architecture at scale and one that offers reliability, security, energy efficiency, mobility, self-organization among others to accommodate these varying IoT service needs.

9. Acknowledgements

We thank all the contributors, reviewers and the valuable comments offered by the chairs to improve this draft.

10. Informative References

[1] Cisco System Inc., CISCO., "Cisco visual networking index: Global mobile data traffic forecast update.", 2016-2021.

[2] Shafig, M., Ji, L., Liu, A., Pang, J., and J. Wang, "A first look at cellular machine-to-machine traffic: large scale measurement and characterization.", Proceedings of the ACM Sigmetrics , 2012.

[3] The European Telecommunications Standards Institute, ETSI., "http://www.etsi.org/.", 1988.

[4] Global Intiative for M2M Standardization, oneM2M., "http://www.onem2m.org/.", 2012.

[5] Constrained RESTful Environments, CoRE., "https://datatracker.ietf.org/wg/core/charter/.", 2013.

[6] Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., and J. Wilcox, "Information-Centric Networking: Seeing the Forest of the Trees.", Hot Topics in Networking , 2011.

[7] Dong, L., Zhang, Y., and D. Raychaudhuri, "Enhance Content Broadcast Efficiency in Routers with Integrated Caching.", Proceedings of the IEEE Symposium on Computers and Communications (ISCC) , 2011.

[8] NSF FIA project, MobilityFirst., "http://mobilityfirst.winlab.rutgers.edu/", 2010.

[9] Thing-to-Thing Research Group, T2TRG., "https://datatracker.ietf.org/rg/t2trg/about/", 2017.

[10] OPC Foundation, OPC., "https://opcfoundation.org/", 2017.

[11] Kim, B., Lee, S., Lee, Y., Hwang, I., and Y. Rhee, "Mobiscape: Middleware Support for Scalable Mobility Pattern Monitoring of Moving Objects in a Large-Scale City.", Journal of Systems and Software, Elsevier, 2011.

[12] Dietrich, D., Bruckne, D., Zucker, G., and P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview", IEEE Transactions on Industrial Electronics, 2010.

[13] Keith, K., Falco, F., and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security", NIST, Technical Report 800-82 Revision 1, 2013.

[14] Darianian, M. and Martin. Michael, "Smart home mobile RFID-based Internet-of-Things systems and services.", IEEE, ICACTE, 2008.

[15] Zhu, Q., Wang, R., Chen, Q., Chen, Y., and W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things", IEEE/IFIP, EUC, 2010.

[16] Biswas, T., Chakrabort, A., Ravindran, R., Zhang, X., and G. Wang, "Contextualized information-centric home network", ACM, Sigcomm, 2013.

[17] Huang, R., Zhang, J., Hu, Y., and J. Yang, "Smart Campus: The Developing Trends of Digital Campus", 2012.

[18] Yan, Y., Qian, Y., Hu, Y., and J. Yang, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges", IEEE Communications Survey and Tutorials, 2013.

[19] Grassi, G., Pesavento, D., Pau, G., Vayyuru, R., Wakikawa, Ryuji., Wakikawa, Ryuji., and Lixia. Zhang, "Vehicular Inter-Networking via Named Data", ACM Hot Mobile (Poster), 2013.

[20] Chai, W., Katsaros, K., Strobbe, M., and P. Romano, "Enabling Smart Grid Applications with ICN", ICN Sigcomm, 2015.

[21] Katsaros, K., Chai, W., Wang, N., and G. Pavlou, "Information-centric Networking for Machine-to-Machine Data Delivery: A Case Study in Smart Grid Applications", IEEE Network, 2014.

[22] Dong, X., "Event-trigger particle filter for smart grids with limited communication bandwidth infrastructure", IEEE Transactions on Smart Grid, 2017.

[23] Mael, A., Maheo, Y., and F. Raimbault, "CoAP over BP for a delay-tolerant internet of things", Future Internet of Things and Cloud (FiCloud), IEEE, 2015.

[24] Patrice, R. and H. Rivano, "Tests Scenario on DTN for IOT III Urbanet collaboration", Dissertation, INRIA, 2015.

[25] Kevin, F., "Comparing Information-Centric and Delay-Tolerant Networking", Local Computer Networks (LCN), 2012 IEEE 37th Conference on. IEEE, 2012..

[26] Miao, Y. and Y. Bu, "Research on the Architecture and Key Technology of Internet of Things (loT) Applied on Smart Grid", IEEE, ICAEE, 2010.

[27] Castro, M. and A. Jara, "An analysis of M2M platforms: challenges and opportunities for the Internet of Things", IMIS, 2012.

[28] Gubbi, J., Buyya, R., and S. Marusic, "Internet of Things (IoT): A vision, architectural elements, and future directions", Future Generation Computer Systems, 2013.

[29] Vandikas, K. and V. Tsiatsis, "Performance Evaluation of an IoT Platform. In Next Generation Mobile Apps, Services and Technologies(NGMAST)", Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014.

[30] Zhang, Y., Yu, R., Nekovee, M., Liu, Y., Xie, S., and S. Gjessing, "Cognitive Machine-to-Machine Communications: Visions and Potentials for the Smart Grid", IEEE, Network, 2012.

[31] Zhou, H., Liu, B., and D. Wang, "Design and Research of Urban Intelligent Transportation System Based on the Internet of Things", Springer Link, 2012.

[32] Alessandrelli, D., Petracca, M., and P. Pagano, "T-Res: enabling reconfigurable in-network processing in IoT-based WSNs.", International Conference on Distributed Computing in Sensor Systems (DCOSS) , 2013.

[33] Kovatsch, M., Mayer, S., and B. Ostermaier, "Moving application logic from the firmware to the Cloud: towards the thin server architecture for the internet of things.", in Proc. 6th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS) , 2012.

[34] Zhang, M., Yu, T., and G. Zhai, "Smart Transport System Based on the Internet of Things", Applied Mechanics and Materials, 2012.

[35] Zhang, A., Yu, R., Nekovee, M., and S. Xie, "The Internet of Things for Ambient Assisted Living", IEEE, ITNG, 2010.

[36] Savola, R., Abie, H., and M. Sihvonen, "Towards metrics-driven adaptive security management in E-health IoT applications.", ACM, BodyNets, 2012.

[37] Jacobson, V., Smetters, D., Plass, M., Stewart, P., Thornton, J., and R. Braynard, "VoCCN: Voice-over Content-Centric Networks", ACM, ReArch, 2009.

[38] Piro, G., Cianci, I., Grieco, L., Boggia, G., and P. Camarda, "Information Centric Services in Smart Cities", ACM, Journal of Systems and Software, 2014.

[39] Gaur, A., Scotney, B., Parr, G., and S. McClean, "Smart City Architecture and its Applications Based on IoT - Smart City Architecture and its Applications Based on IoT", Procedia Computer Science, Volume 52, 2015, Pages 1089-1094.

[40] Herrera-Quintero, L., Banse, K., Vega-Alfonso, J., and A. Venegas-Sanchez, "Smart ITS sensor for the transportation planning using the IoT and Bigdata approaches to produce ITS cloud services", 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, 2016, pp. 1-7.

[41] Melis, A., Pardini, M., Sartori, L., and F. Callegati, "Public Transportation, IoT, Trust and Urban Habits", Internet Science: Third International Conference, INSCI 2016, Florence, Italy, September 12-14, 2016, Proceedings.

[42] Tonneau, A., Mitton, N., and J. Vandaele, "A Survey on (mobile) Wireless Sensor Network Experimentation Testbeds", 2014 IEEE International Conference on Distributed Computing in Sensor Systems, Marina Del Rey, CA, 2014, pp. 263-268.

[43] Zhilin, Y., "Mobile phone location determination and its impact on intelligent transportation systems", IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 1, pp. 55-64, Mar 2000.

[44] Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation", IEEE Communications Magazine, vol. 47, no. 11, pp. 84-95, November 2009.

[45] Zhang, Yu., Afanasyev, A., Burke, J., and L. Zhang, "A survey of mobility support in named data networking", Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on. IEEE, 2016.

[46] Xylomenos, G., Ververidis, C., Siris, V., and N. Fotiou et al, "A survey of information-centric networking research", IEEE Communications Surveys and Tutorials, Volume: 16, Issue: 2, Second Quarter 2014 .

[47]   Mavromoustakis, C., Mastorakis, G., and J. Batalla, "Internet of Things (IoT) in 5G Mobile Technologies", ISBN,3319309137,Springer.

[48]   Firner, S., Medhekar, S., and Y. Zhang, "PIP Tags: Hardware Design and Power Optimization", in Proceedings of HotEmNets, 2008.

[49]   Masek, P., Masek, J., Frantik, P., and R. Fujdiak, "A Harmonized Perspective on Transportation Management in Smart Cities: The Novel IoT-Driven Environment for Road Traffic Modeling", Sensors, Volume 16, Issue 11, 2016.

[50]   Abreu, D., Velasquez, K., Curado, M., and E. Monteiro, "A resilient Internet of Things architecture for smart cities", Annals of Telecommunications, Volume 72, Issue 1, Pages 19-30, 2017.

[51]   Ravindran, R., Biswas, T., Zhang, X., Chakrabort, A., and G. Wang, "Information-centric Networking based Homenet", IEEE/IFIP, 2013.

[52]   Dannewitz, C., D' Ambrosio, M., and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks", 2013.

[53]   Fasoloy, E., Rossiy, M., and M. Zorziy, "In-network Aggregation Techniques for Wireless Sensor Networks: A Survey", IEEE Wireless Communications, 2007.

[54]   Chai, W., He, D., and I. Psaras, "Cache "less for more" in information-centric networks", ACM, IFIP, 2012.

[55]   Eum, S., Nakauchi, K., Murata, M., Shoji, Yozo., and N. Nishinaga, "Catt: potential based routing with content caching for icn", IEEE Communication Magazine, 2012.

[56]   Drira, W. and F. Filali, "Catt: An NDN Query Mechanism for Efficient V2X Data Collection", Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking Workshops (SECON Workshops), 2014.

[57]   Eum, S., Shvartzshnaider, Y., Francisco, J., Martini, R., and D. Raychaudhuri, "Enabling internet-of-things services in the mobilityfirst future internet architecture", IEEE, WoWMoM, 2012.

[58] Raychaudhuri, D., Nagaraj, K., and A. Venkatramani, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet.", ACM SIGMOBILE Mobile Computing and Communications Review 16.3 (2012): 2-13.

[59] Sun, Y., Qiao, X., Cheng, B., and J. Chen, "A low-delay, lightweight publish/subscribe architecture for delay-sensitive IOT services", IEEE, ICWS, 2013.

[60] Azgin, A., Ravindran, R., and GQ. Wang, "Mobility study for Named Data Networking in wireless access networks", IEEE, ICC, 2014.

[61] Azgin, A., Ravindran, R., Chakraborti, A., and GQ. Wang, "Seamless Producer Mobility as a Service in Information Centric Networks", ACM ICN Sigcomm, IC5G Workshop, 2016.

[62] Wang, L., Wakikawa, R., Kuntz, R., and R. Vuyyuru, "Data Naming in Vehicle-to-Vehicle Communications", IEEE, Infocm, Nomen Workshop, 2012.

[63] Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T., and M. Wahlisch, "Information Centric Networking in the IoT:Experiments with NDN in the Wild", ACM, ICN Siggcomm, 2014.

[64] Ascigil, O., Rene, S., Xylomenos, G., Psaras, I., and G. Pavlou, "A Keyword-based ICN-IoT Platform", ACM, ICN Sigcomm, 2017..

[65] Simona, C. and M. Mongiello, "Pushing the role of information in ICN", Telecommunications (ICT), 2016 23rd International Conference on. IEEE, 2016..

[66] Li, B., Huang, D., Wang, Z., and Y. Zhu, "Attribute-based Access Control for ICN Naming Scheme", IEEE Transactions on Dependable and Secure Computing, vol.PP, no.99, pp.1-1..

[67] Polyzos, G. and N. Fotiou, "Building a reliable Internet of Things using Information-Centric Networking", Journal of Reliable Intelligent Environments, vol.1, no.1, 2015..

[68] Pandurang, K., Xu, W., Trappe, W., and Y. Zhang, "Temporal privacy in wireless sensor networks: Theory and practice", ACM Transactions on Sensor Networks (TOSN) 5, no. 4 (2009): 28..

[69] Trossen, D., Sarela, M., and K. Sollins, "Arguments for an information-centric internetworking architecture.", ACM SIGCOMM Computer Communication Review 40.2 (2010): 26-33.

[70] Zhang, G., Li, Y., and T. Lin, "Caching in information centric networking: A survey.", Computer Networks 57.16 (2013): 3128-3141.

[71] Gronbaek, I., "Architecture for the Internet of Things (IoT): API and interconnect", IEEE, SENSORCOMM, 2008.

[72] Tian, Y., Liu, Y., Yan, Z., Wu, S., and H. Li, "RNS-A Public Resource Name Service Platform for the Internet of Things", IEEE, GreenCom, 2012.

[73] Roussos, G. and P. Chartier, "Scalable id/locator resolution for the iot", IEEE, iThings,CPSCom, 2011.

[74] Amadeo, M. and C. Campolo, "Potential of information-centric wireless sensor and actuator networking", IEEE, ComManTel, 2013.

[75] Nelson, S., Bhanage, G., and D. Raychaudhuri, "GSTAR: generalized storage-aware routing for mobilityfirst in the future mobile internet", ACM, MobiArch, 2011.

[76] Trappe, W., Zhang, Y., and B. Nath, "MIAMI: methods and infrastructure for the assurance of measurement information", ACM, DMSN, 2005.

[77] Rouf, I., Mustafa, H., Taylor, T., Oh, S., Xu, W., Gruteser, M., Trappe, W., and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study", USENIX, 2010.

[78] Liu, R. and W. Trappe, "Securing Wireless Communications at the Physical Layer", Springer, 2010.

[79] Xiao, L., Greenstein, L., Mandayam, N., and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels", IEEE Transactions on Wireless Communications, 2008.

[80] Sun, S., Lannom, L., and B. Boesch, "Handle system overview", IETF, RFC3650, 2003.

[81] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[82] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[83] Barnes, R., "Use Cases and Requirements for JSON Object Signing and Encryption (JOSE)", RFC 7165, DOI 10.17487/RFC7165, April 2014, <https://www.rfc-editor.org/info/rfc7165>.

[84] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[85] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <https://www.rfc-editor.org/info/rfc3561>.

[86] Mosko, M., Uzun, E., and C. Wood, "CCNx Key Exchange Protocol Version 1.0", draft-wood-icnrg-ccnxkeyexchange-02 (work in progress), July 2017.

[87] Sun, S., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", 2014.

[88] Liu, X., Trappe, W., and Y. Zhang, "Secure Name Resolution for Identifier-to-Locator Mappings in the Global Internet", IEEE, ICCCN, 2013.

[89] Boguna, M., Fragkiskos, P., and K. Dmitri, "Sustaining the internet with hyperbolic mapping", Nature Communications, 2010.

[90] Shang, W., "Securing building management systems using named data networking", IEEE Network 2014.

[91] Fayazbakhsh, S. and et al, "Less pain, most of the gain: Incrementally deployable icn", ACM, Siggcomm, 2013.

[92] Burke, J. and et. et al, "Securing instrumented environments over Content-Centric Networking: the case of lighting control", INFOCOM, Computer Communications Workshop, 2013.

[93] Rao, A., Schelen, O., and A. Lindgren, "Performance Implications for IoT over Information Centric Networks", Performance Implications for IoT over Information Centric Networks, ACM CHANTS 2016.

[94] Hahm, O., Baccelli, E., Schmidt, T., Wahlisch, M., Adjih, C., and L. Massoulie, "Low-power Internet of Things with NDN and Cooperative Caching", ICN, Sigcomm, 2017.

[95] Li, S., Zhang, Y., Dipankar, R., and R. Ravindran, "A comparative study of MobilityFirst and NDN based ICN-IoT architectures", IEEE, QShine, 2014.

[96] Chen, J., Li, S., Yu, H., Zhang, Y., and R. Ravindran, "Exploiting icn for realizing service-oriented communication in iot", IEEE, Communication Magazine, 2016.

[97] Quevedo, J., Corujo, D., and R. Aguiar, "A Case for ICN usage in IoT environments", Global Communications Conference GLOBECOM, IEEE, Dec 2014, Pages 2770-2775.

[98] Lindgren, A., Ben Abdesslem, F., Ahlgren, B., and O. Schelen, "Design Choices for the IoT in Information-Centric Networks", IEEE Annual Consumer Communications and Networking Conference (CCNC) 2016.

[99] Grieco, L., Alaya, M., and K. Drira, "Architecting Information Centric ETSI-M2M systems", IEEE, Pervasive and Computer Communications Workshop (PERCOM), 2014.

[100] Compagno, A., Conti, M., and R. Dorms, "OnboardICNg: a Secure Protocol for On-boarding IoT Devices in ICN", ICN, Sigcomm, 2016.

[101] Grieco, L., Rizzo, A., Colucci, R., Sicari, S., Piro, G., Di Paola, D., and G. Boggia, "IoT-aided robotics applications: technological implications, target domains and open issues", Elsevier Computer Communications, Volume 54, 1 December, 2014.

[102] InterDigital, WhitePaper., "Standardized M2M Software Development Platform", 2011.

[103] Boswarthick, D., "M2M Communications: A Systems Approach", 2012.

[104] Swetina, J., Lu, G., Jacobs, P., Ennesser, F., and J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M", IEEE Wireless Communications, Volume 21, Number 3, June 2014.

[105] Wang, L., Wang, Z., and R. Yang, "Intelligent Multiagent Control System for Energy and Comfort Management in Smart and Sustainable Buildings", IEEE Transactions on Smart Grid, vol. 3, no. 2, pp. 605-617, June 2012..

[106] Lawrence, T., Boudreau, M., and L. Helsen, "Ten questions concerning integrating smart buildings into the smart grid, Building and Environment", Building and Environment, Volume 108, 1 November 2016, Pages 273-283..

[107] Hassan, A. and D. Kim, "Named data networking-based smart home", ICT Express 2.3 (2016): 130-134..

[108] Burke, J., Horn, A., and A. Marianantoni, "Authenticated lighting control using named data networking", UCLA, NDN Technical Report NDN-0011 (2012)..

[109] Afanasyev, A., "Packet fragmentation in ndn: Why ndn uses hop-by-hop fragmentation.", UCLA, NDN Technical Report NDN-0032 (2015)..

[110] Quan, Wei., Xu, C., Guan, J., Zhang, H., and L. Grieco, "Scalable Name Lookup with Adaptive Prefix Bloom Filter for Named Data Networking", IEEE Communications Letters, 2014.

[111] Wang, Yi., Pan, T., Mi, Z., Dai, H., Guo, X., Zhang, T., Liu, B., and Q. Dong, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage Bloom filters", INFOCOM, 2013.

[112] So, W., Narayanan, A., Oran, D., and Y. Wang, "Toward fast NDN software forwarding lookup engine based on Hash tables", ACM, ANCS, 2012.

[113] Amadeo, M., Campolo, C., Iera, A., and A. Molinaro, "Named data networking for IoT: An architectural perspective", IEEE, EuCNC, 2014.

[114] Amadeo, M., Campolo, C., Iera, A., and A. Molinaro, "Information centric networking in iot scenarios: The case of a smart home", IEEE ICC, June 2015.

[115] Somani, N., Chanda, A., Nelson, S., and D. Raychaudhuri, "Storage- Aware Routing for Robust and Efficient Services in the Future Mobile Internet", Proceedings of ICC FutureNet V, 2012.

[116] Blefari Melazzi, N., Detti, A., Arumaithurai, M., and K. Ramakrishnan, "Internames: A name-to-name principle for the future internet", QShine, August 2014.

[117] Sifalakis, M., Kohler, B., Christopher, C., and C. Tschudin, "An information centric network for computing the distribution of computations", ACM, ICN Sigcomm, 2014.

[118] Lu, R., Lin, X., Zhu, H., and X. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots", INFOCOM, 2009.

[119] Wang, H. and W. He, "A reservation-based smart parking system", The First International Workshop on Cyber-Physical Networking Systems, 2011.

[120] Qian, L., "Constructing Smart Campus Based on the Cloud Computing and the Internet of Things", Computer Science 2011.

[121] Project, BonVoyage., "European Unions - Horizon 2020, http://bonvoyage2020.eu/", 2016.

[122] Li, S., Zhang, Y., Raychaudhuri, D., Ravindran, R., Zheng, Q., Wang, GQ., and L. Dong, "IoT Middleware over Information-Centric Network", Global Communications Conference (GLOBECOM) ICN Workshop, 2015.

[123] Li, S., Chen, J., Yu, H., Zhang, Y., Raychaudhuri, D., Ravindran, R., Gao, H., Dong, L., Wang, GQ., and H. Liu, "MF-IoT: A MobilityFirst-Based Internet of Things Architecture with Global Reachability and Communication Diversity", IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI), 2016.

[124] Adhatarao, S., Chen, J., Arumaithurai, M., and X. Fu, "Comparison of naming schema in ICN", IEEE LANMAN, June , 2016.

[125] Campolo, C., Corujo, D., Iera, A., and R. Aguiar, "Information-centric Networking for Internet-of-things: Challenges and Opportunities", IEEE Networks, Jan , 2015.

[126] Hussain, R., Bouk, S., Javaid, N., and Adil. Khan, "Realization of VANET-Based Cloud Services through Named Data Networking", IEEE Communication Magazine, 2018.

[127] Sobia, A. and et al., "Hierarchical and Flat-Based Hybrid Naming Scheme in Content-Centric Networks of Things", IEEE Internet of Things Journal, 2018.

[128] Sobia, A. and et al., "Towards Information-Centric Networking (ICN) naming for Internet of Things (IoT): the case of smart campus.", Proceedings of the International Conference on Future Networks and Distributed Systems. ACM, 2017.

[129] Agnese, V. and et al., "Publish subscribe in mobile information centric networks: Modeling and performance evaluation.", Computer Networks, 2017.

Authors' Addresses

Ravishankar Ravindran
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ravi.ravindran@huawei.com

Yanyong Zhang
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: yyzhang@winlab.rutgers.edu

Luigi Alfredo Grieco
Politecnico di Bari (DEI)
Via Orabona 4
Bari 70125
Italy

Email: alfredo.grieco@poliba.it

Anders Lindgren
RISE SICS
Box 1263
Kista SE-164 29
SE

Email: anders.lindgren@ri.se

Jeff Burke
UCLA REMAP
102 East Melnitz Hall
Los Angeles, CA 90095
USA

Email: jburke@ucla.edu

Bengt Ahlgren
RISE SICS
Box 1263
Kista, CA SE-164 29
SE

Email: bengt.ahlgren@ri.se

Aytac Azgin
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: aytac.azgin@huawei.com