ICN Research Group Y. Zhang
Internet-Draft D. Raychadhuri
Intended status: Informational WINLAB, Rutgers University
Expires: December 28, 2017 L. Grieco
Politecnico di Bari (DEI)
E. Baccelli
INRIA
J. Burke
UCLA REMAP
R. Ravindran
G. Wang
Huawei Technologies
A. Lindgren
B. Ahlgren
RISE SICS
O. Schelen
Lulea University of Technology
June 26, 2017

Design Considerations for Applying ICN to IoT
draft-zhang-icnrg-icniot-01

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 over a horizontal unified, defragmented IoT architecture. Such an architecture will make objects 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 todays Internet and mostly information-centric nature of the IoT system. To address this mismatch, an information-centric network (ICN) architecture can provide a common set of protocols and services, called 'ICN-IoT', which can be used to build IoT

platforms.
   ICN-IoT leverages the salient features of ICN, and thus provides
   naming, security, mobility support,scalability, and efficient
content
   and service delivery.

   This draft summarizes general IoT demands, and covers the challenges
   and design considerations ICN faces to realize a ICN-IoT framework
   based on ICN architecture.

Status of This Memo

Copyright Notice

Table of Contents

1. IoT Motivation

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 towards a more unified IoT architecture, in which a large number of objects connect to the Internet, available for interactions among themselves, as well as interactions with many different applications across boundaries of administration and domains. General IoT applications involve sensing, processing, and secure content distribution occurring at various timescales and at multiple levels of hierarchy depending on the application requirements. This requires the system to adopt a unified architecture providing pull, push and publish/subscribe mechanisms using application abstractions, common naming, payload, encryption and signature schemes. This requires open APIs to be generic enough to support commonly used interactions between consumers, content producer, and IoT services, as opposed to proprietary APIs that are common in today's systems. Building a unified IoT architecture, however, poses great challenges on the underlying network and systems. To name a few, it needs to support 50-100 Billion networked objects [1], many of which are mobile. The objects will have

extremely heterogeneous means of connecting to the Internet, often with severe resource constraints. Interactions between the applications and objects are often real-time and dynamic, requiring strong security and privacy protections. In addition, many IoT applications are inherently information centric (e.g., data consumers usually need data sensed from the environment without any reference to the sub-set of sensors that will provide the asked information).

Taking a general IoT perspective, we first motivate the discussion of ICN for IoT using well known scenarios. Then we discuss the IoT requirements generally applicable to many well known IoT scenarios. We then discuss how the current application-layer unified IoT architectures fail to meet these requirements. We follow this by key ICN features that makes it a better candidate to realize an unified IoT framework. We then discuss IoT design challenges from an ICN perspective and requirements posed towards its design.

2. Motivating ICN for IoT

ICN offers many features including name-based networking, content object security, caching, computing and storage, mobility, context-aware networking (see Section 3.6) and support for ad hoc networking features, all of which have to be realized in an application-specific means in the context of IP-IoT. These compelling features enable a distributed and intelligent data distribution platform to support heterogeneous IoT services with features like device bootstrapping with minimal configuration, simpler protocols to aid self-organizing among the IoT elements, natural support for compute and caching logic at strategic points in the network. We discuss these features through the following scenarios that are difficult to realize over IP today, and whose characteristics we argue match the features offered by ICN.

- Smart Mobility: Smarter end-user devices and Machine-to-Machine (M2M) connection are undergoing a significant growth. By 2021, there will be more than 10 billion mobile devices and connection, including smartphones, tablets, wearables, vehicles [1]. Involved fields range from medical and healthcare, fitness, clothing, to environmental monitoring [40]. In particular, one of the most affected domain is transportation and the so-called Intelligent Transport Systems (ITS) [42]. It aims at providing multi-modal transportation, embracing public and private municipal, regional, national, trans-national vehicles and fleets. This extremely heterogeneous eco-system of means of transportation is made available to users and citizens through advanced services. These

services are able to fulfill usability requirements while pursuing
system level objectives, thus including: (i) the reduction of the
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 [117]. In this context, IoT technologies can play a pivotal, in particular, Traffic Management Systems (TMS) aided by IoT technologies are creating novel approaches to traffic modeling [47]. Moreover, such features enable advanced design paradigms (e.g., Mobility as a Service (MaaS) [39]) with huge implications in systems architectures [48]. As a consequence, smart mobility support can be a significant use case of ICN-IoT. The important ICN features that corroborate mobility support are:

* The location independence of content allows one to manage consumer mobility in a simpler way than IP. Different from Mobile IP, that needs 'triangular routing' to locate moving hosts, ICN envisions that the consumers just needs to re-issue content requests after changing the attachment point [43];

* Since content is not bound to a specific location, it can be cached anywhere in the network. This caching mechanism adds redundancy to the system. Therefore, if the producer loses connectivity while it is moving, a content request can be resolved to an intermediate node en-route or routed towards a caching node [43];

* The content request-response communication paradigm decouples publications and subscriptions in time and space. Therefore, entities involved are not aware of each other and do not need to be connected at the same time [44];

* The use of in-network Name Resolution Service design allows to identify content name's current location in the network, thanks to its network function of updating named entity location information [56].

From a technological perspective, open challenges are: (i) interoperability across different IoT technologies; (ii) namespace design able to harmonize ITS standards; (iii) scalable data-sharing model across real-time (and non real-time) traffic sources; (iv) definition of travel-centric services based on ICN-IoT; (v) seamless support to mobility; (vi) content authentication and cryptography.

- Smart Building: Buildings are gaining smart capabilities that allow to enhance comfort, provide safety and security, manage efficiently energy [101]. In particular, smart buildings are no longer simple energy consumer, but can also be prosumers with on-site energy generation systems. These systems can improve

building's usability towards: (i) Smart heating, ventilation, and air conditioning (HVAC), (ii) Smart lightings, (iii) Plug loads, (iv) Smart windows. The main requirements of those sub-systems are [101]: (i) context awareness; (ii) resource-constrained devices; (iii) interoperability across heterogeneous technologies;
(iv) security and privacy protection. The ICN paradigm could ease
the fulfillment of those requirements because, usually, smart building services are information centric by design: this means that every time an autonomic management loop is established within
the smart building to control some physical variables of interest,
the information exchanged between users, sensors, actuators, and controllers do not immediately translate to specific nodes within the building but could be provided by multiple sensors / gateways.
The relevance of ICN in Smart Building is recognized in literature
with reference to the several frameworks deployed in various environments. For instance, in [61], nodes are distributed in different rooms, floors, and buildings of a campus university and their energy consumption and individual behavior are monitored. Smart home application is investigated in [103], by evaluating data retrieval delay and data packet loss. Moreover, [104] designed and tested lighting control over NDN in a theater. In this context, specific ICN challenges are: (i) design of a scalable namespace for uniquely identifying the information of interest, (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 Grids are increasingly transforming into cyber-physical systems [18] with the goal of maximum automation towards efficiency and minimal human intervention. The system is very complex comprising of power distribution grids, end user applications (e.g. EV charging systems, appliances etc), smart monitoring systems (spanning end user and the power grids), heterogeneous energy producing sources (including prosumers), and load distribution and balancing systems. Current smart grid systems are managed using Supservisory Control and Data Acquisition (SCADA) frameworks that are centralized and highly restrictive unidirectional communication support [19]. Hence the requirement is towards : 1) greater flexibility to distribute the energy from the feeder through real-time reconfiguration of multiple monitoring devices (e.g. phasor measurement units (PMUs)), and management operations which require efficient data delivery infrastructure; 2) large scale data delivery infrastructure, which also include latency sensitive

applications,
inter-connecting heterogeneous smart grid producing, monitoring and consuming end user devices; 3) Resiliency,which is critical to

the operation and protection of the grid; 4)Security, to protect mission critical grid applications from network intrusions ; 5) understanding machine-to-machine traffic patterns for optimal placement of storage and computing for maximum efficiency. Smart grids can benefit from ICN in the following ways [20] :

* Smart grid will benefit from naming content than hosts, as it is more likely that data generated by one subsystem will be useful for multiple entities. Further, naming content allows to enable many-to-many model of communication, which is very in-efficient in host-centric architectures.

* ICN features of in-network computing, storage and caching will enable better use of network resources and benefit diverse application needs varying from applications that has low bitrate and is latency tolerant (e.g. smart grid and energy pricing) to higher data rate ones with stringent delay/ disruption requirements (e.g. synchrophasor measurements). Also it is typical in smart grid systems to have applications consuming the same data at different rates in which case in-network caching and computing could help.

* Host-centric networking exposes a mission critical infrastructure like smart grid infrastructure to intrusion and DOS attacks, this is directly related to exposing the IP addresses of critical applications and subsystems. Naming content, service or device de-couples it from the location, reducing the exposure to target a specific smart grid subsystem based on a geographical context.

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

o Smart Industrial Automation : In a smart and connected industry environment, there is a multitude of equipment with sensors that generate large volumes of data during normal operation. This range from highly time-critical data for real-time control of production processes, to less time-critical data that is collected to central cloud environment for control room monitoring, to pure log data without latency requirements that is mainly kept for a posteriori analysis. Industrial wireless networks are harsh environments with lots of potential interference at the same time as hard reliability and real-time requirements are placed by many applications. This means that available network capacity is not always high, so congestion is likely to be experienced by traffic with less stringent timing requirements. One such example is when

errors occur in the production process, a mobile workforce will need to investigate the problem on-site and will need high resolution data from the faulty machine as well as other process data from other parts of the plant. The mobile workforce will locally perform diagnostics or maintenance and they rely on the information from the production system both for safety and to solve any issues in the plant. They rely on both historical data in order to pinpoint the root cause of the problems, as well as the current data flows in order to assess the present state of the equipment under control. High resolution measurements are generated close to the mobile workforce while the historic data has to be retrieved from the historian servers. Multiple workers involved in the process will access the same data, possibly with a slight time-shift. The network thus need to support a mobile users to get access to data flows in a way suitable for their physical location and task requirements. Introducing ICN functionality into the system can introduce several benefits that will 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 OPC Foundation [10] maps well to the CCN/NDN name space.

* ICN provides the possibility to get newest data without knowing the location of the caches or whether a particular piece of data is available locally or in a central repository. Also gives the possibility to get either local high-resolution data or remote low-resolution data (no need to store all data centrally, which is maybe not even possible due to large data volumes). May require known naming conventions or routing policies that can route interests to the right location.

* Reduces network usage 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 is inherently supported without the need to maintain connection state.

* Removing tedious configurations in clients since that is provided by the infrastructure.

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

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

3. IoT Architectural Requirements

A unified IoT platform has to support interactions among a large number of mobile devices across the boundaries of organizations and domains. As a result, it naturally poses stringent requirements in every aspect of the system design. Below, we outline a few important requirements that a unified 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 each device, data items generated by these devices, or a group of devices towards a common objective. Naming has the following requirements. Firstly, names need to be persistent against dynamic features that are common in IoT systems, such as lifetime, mobility or migration. Secondly, names that are derived from the keys need to be self-certifying, for both device-centric communication and content-centric communication. For device-centric communication, the binding between device names and the device must be secure. For content-centric communication, the binding between the names and the content has to be secure. Thirdly, names usually serve multiple purposes: routing, security (self-certifying) or human-readability. For IoT applications, the choice of flat versus human readable names needs to be made considering application and network requirements such as privacy and network level scalability, and the name space explosion that may occur because of complex relationship between name hierarchies [120] which might confound application logic. In order to ensure the trustworthiness of the names, a name certificate service (NCS) needs to be considered. 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's end. In short, the NCS must provide services analogous to those provided by a Public Key Infrastructure (PKI). In ICN, users may either generate their own public keys and submit them to the NCS for registration, or may contact the NCS to acquire public keys. Consequently, the NCS publishes approved cryptographic suites, object categories and object description formats, as well as allows users to self-certify themselves.

3.2. Security and Privacy

A variety of security and privacy concerns exist in IoT. For example the unified IoT architecture makes physical objects accessible to applications across organizations and domains. Further, it often integrates with critical infrastructure and industrial systems with life safety implications, bringing with it significant security challenges and regulatory requirements [13], as will be discussed in Section 6.3. 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: (1) privacy of data producer/consumer that is directly related to each individual vertical domain such as heath, electricity, etc., (2) privacy of data content, and (3) privacy of contextual information such as time and location of data transmission [65].

3.3. Scalability

Cisco predicts there will be around 50 Billion IoT devices such as sensors, RFID tags, and actuators, on the Internet by 2020 [1]. As mentioned above, a unified IoT platform needs to name every entity such as data, device, service etc. Scalability has to be addressed at multiple levels of the IoT architecture including naming, security, name resolution, routing and forwarding level. Mobility adds further challenge 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. In addition scalability is also affected because of IoT system specific features such as IoT resource object count, state and rate of information updates 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 the most resource-constrained and type 3 the most resource-rich [45]. In general, there are the following types of resources: 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 [46]. Flexible techniques to collect the relevant information are required, and uploading every single produced data to a central server is undesirable. Computing constraints limit the type and amount of processing these devices can perform. As a result, more complex

processing needs to be conducted in cloud servers or at opportunistic points, example at the network edge, hence it is important to balance local computation versus communication cost.

Storage constraints of the IoT devices limit the amount of data that can be stored on the devices. This constraint means that unused sensor data may need to be discarded or stored in aggregated compact form time to time. Bandwidth constraints of the IoT devices limit the amount of communication. Such devices will have the same implication on the system architecture as with the power constraints; namely, we cannot afford to collect single sensor data generated by the device and/or use complex signaling protocols. It is also worth mentioning that idle chatter in the background is strongly discouraged to maintain connectivity or other volatile state.

User interface constraints refer to whether the device is itself capable of directly interacting with a user should the need arise (e.g., via a display and keypad or LED indicators) or requires the network connectivity, either global or local, to interact with humans.

The above discussed device constraints also affect application performance with respect to latency.

3.5. Traffic Characteristics

IoT traffic can be broadly classified into local area traffic and wide area traffic. Local area traffic is among nearby devices. For example, neighboring cars may work together to detect potential hazards on the highway, sensors deployed in the same room may collaborate to determine how to adjust the heating level in the room. These local area communications often involve data aggregation and filtering, have real time constraints, and require fast device/data/service discovery and association. At the same time, the IoT platform has to also support wide area communications. For example, in Intelligent Transportation Systems, re-routing operations may require a broad knowledge of the status of the system, traffic load, availability of freights, whether forecasts and so on. Wide area communications require efficient data/service discovery and resolution 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 volume in some applications such as [2], which means that we have to optimize the bandwidth/energy consumption in both directions. Further, IoT traffic demonstrates certain periodicity and burstiness [2]. As a

result, when provisioning the system, the shape of the traffic volume has to be properly accounted for.

3.6. Contextual Communication

Many IoT applications rely on dynamic contexts in the IoT system to initiate, maintain and terminate communication among 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 in the same location. There are two types of contexts: long-term quasi static contexts and short-term dynamic contexts. In this draft, we focus on the latter, which are more challenging to support, requiring fast formation, update, lookup and association For example, cars traveling on the highway may form a "cluster" based upon their temporal physical proximity as well as the detection of the same event. These temporary groups are referred to as contexts. IoT applications need to support interactions among the members of a context, as well as interactions across contexts.

Temporal context can be broadly categorized into two classes, long-term contexts such as those that are based upon social contacts as well as stationary physical locations (e.g., sensors in a car/ building), and short-term contexts such as those that are based upon temporary proximity (e.g., all taxicabs within half a mile of the Time Square at noon on Oct 1, 2013). Between these two classes, short-term contexts are more challenging to support, requiring fast formation, update, lookup and association.

3.7. Handling Mobility

There are several degrees of mobility in a unified IoT architecture, ranging from static as in fixed assets to highly dynamic in vehicle-to-vehicle environments.

Mobility in the IoT architecture can mean 1) the data producer mobility (i.e., location change), 2) the data consumer mobility, 3) IoT Network mobility (e.g., a body-area network in motion as a person is walking); and 4) disconnection between the data source and destination pair (e.g., due to unreliable wireless links). The requirement on mobility support is to be able to deliver IoT data below an application's acceptable delay constraint in all of the above cases, and if necessary to negotiate different connectivity or security constraints specific to each mobile context. More detailed discussions are presented in Section 6.7.

3.8. Storage and Caching

Storage and caching plays a very significant role depending on the type of IoT ecosystem, also a function subjected to privacy and security guidelines. Caching is usually done for increasing data availability in the network and reliability purposes, especially in wireless scenarios in the network access. Storage is more important for IoT, storing data for long term analysis. Data is stored in strategic locations in the network to reduce control and computation overhead. In a unified IoT architecture, depending on application requirements, content caching will be strictly driven by application level policies considering privacy requirements. If for certain kind of IoT data pervasive caching is allowed, intermediate nodes don't need to always forward a content request to its original creator; rather, receiving a cached copy is sufficient for IoT applications. This optimization may greatly reduce the content access latencies.

Furthermore considering hierarchical nature of IoT systems, ICN architectures enable flexible heterogeneous and potentially fault-tolerant approach to storage providing persistence at multiple levels.

Hence in the context of IoT while ICN allows resolution to replicated stored copies, it should also strive for the balance between content security/privacy and regulations considering application requirements.

3.9. Communication Reliability

IoT applications can be broadly categorized into mission critical and non-mission critical. 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 probability of error during information transfer. To summarize, reliable communication desires the following capabilities for the underlying system: (1) seamless mobility support under normal operating conditions, (2) efficient routing in the presence of intermittent disconnection, (3) QoS aware routing, (4) support for redundancy at all levels of a system (device, service, network, storage etc.), and (5) 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

The unified IoT architecture should be able to self-organize to meet various application requirements, especially the capability to quickly discover heterogeneous and relevant (local or global) devices/data/services based on the context. This discovery can be achieved through an efficient publish-subscribe service, or through private community grouping/clustering based upon trust and other security requirements. In the former case, the publish-subscribe service must be efficiently implemented, able to support seamless mobility, in- network caching, name-based routing, etc. In the latter case, the IoT architecture needs to discover the private community groups/clusters efficiently.

Another aspect of self-organization is decoupling the sensing Infrastructure from applications. In a unified IoT architecture, various applications run on top of a vast number of IoT devices. Upgrading the firmware of the IoT devices is a difficult work. It is also not practical to reprogram the IoT devices to accommodate every change of the applications. The infrastructure and the application specific logics need to be decoupled. A common interface is required to dynamically configure the interactions between the IoT devices and easily modify the application logics on top of the sensing/actuating infrastructure [30] [31].

3.11. Ad hoc and Infrastructure Mode

Depending upon whether there is communication infrastructure, an IoT system can operate either in ad-hoc or infrastructure mode.

For example, a vehicle may determine to report its location and status information to a server periodically through cellular connection, or, a group of vehicles may form an ad-hoc network that collectively detect road conditions around them. In the cases where infrastructure is unavailable, one of the participating nodes may choose to become the temporary gateway.

The unified IoT architecture needs to design a common protocol that serves both modes. Such a protocol should address the challenges that arise in these two modes: (1) scalability and low latency for the infrastructure mode and (2) 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

An IoT platforms' service, control and data plane will be governed by its own management infrastructure which includes distributed and centralized middleware, discovery, naming, self-configuring, analytic functions, and information dissemination to achieve specific IoT system objectives [25][26][27]. Towards this, new IoT management mechanisms and service metrics need to be developed to measure the success of an IoTdeployment. Considering an IoT systems' defining characteristics such as, its potential large number of IoT devices, objective to save power, mobility, and ad hoc communication, autonomic self-management mechanisms become very critical. Further considering its hierarchical information processing deployment model, the platform needs to orchestrate computational tasks according to the involved sensors and the available computation resources which may change over time. An efficient computation 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, towards a unified IoT architecture in which the existing silo IoT systems, as well as new systems that are rapidly deployed. By unified, we mean all the application and network components that use common APIs to interact with each other. This will make their data and services accessible to general Internet applications (as in ETSI- M2M and oneM2M standards). In such a unified architecture, resources can be accessed over Internet and shared across the physical boundaries of the enterprise. However, current approaches to achieve this objective are mostly based upon service overlays over the Internet, whose inherent inefficiencies due to IP protocol [56] hinders the architecture from satisfying the IoT requirements outlined earlier, particularly in terms of scalability, security, mobility, and self-organization, discussed more in Section 4.2.

4.1. Silo IoT Architecture

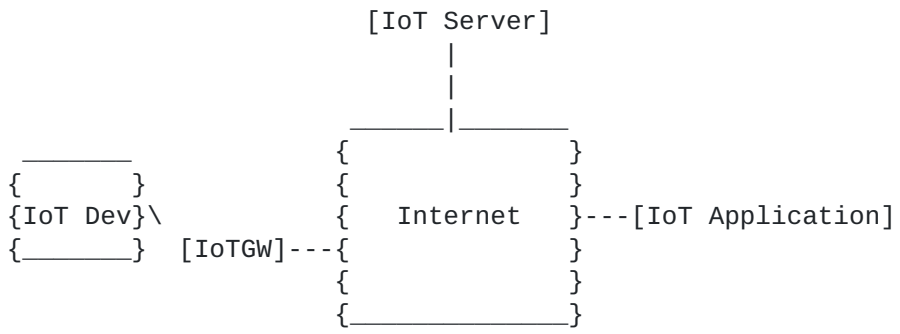

Figure 1:Silo architecture of standalone IoT systems

A typical standalone IoT system is illustrated in Figure 1, which includes devices, a gateway, a server and applications. Many IoT devices have limited power and computing resources, unable to directly run normal IP access network (Ethernet, WIFI, 3G/LTE etc.) protocols. Therefore they use the IoT gateway to the server. Through the IoT server, applications can subscribe to data collected by devices, or interact with devices.

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 the device-level abstraction, instead of information driven, which may sometimes lead to a fragmented protocol space that requires a higher-level solution for better interoperability.

4.2. Application-Layer Unified IoT Solutions

The current approach to a unified IoT architecture is to make IoT gateways and servers adopt standard APIs. IoT devices connect to the Internet through the standard APIs and IoT applications subscribe and receive data through standard control and data APIs. Building 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], oneM2M[4]. Network operators can use frameworks to build common IOT gateways and servers for their customers. In addition, IETF's CORE working group [5] is developing a set of protocols like CoAP (Constrained Application Protocol) [78], that is a lightweight protocol modeled after HTTP [79] and adapted specifically for the Internet of Things (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 Machine-to-Machine communications, as well as IoT. For example, oneM2M (which is one of leading standards for unified M2M architecture) has both the protocol bindings to HTTP and CoAP for its primitives. Figure 2 shows the architecture adopted in this approach.

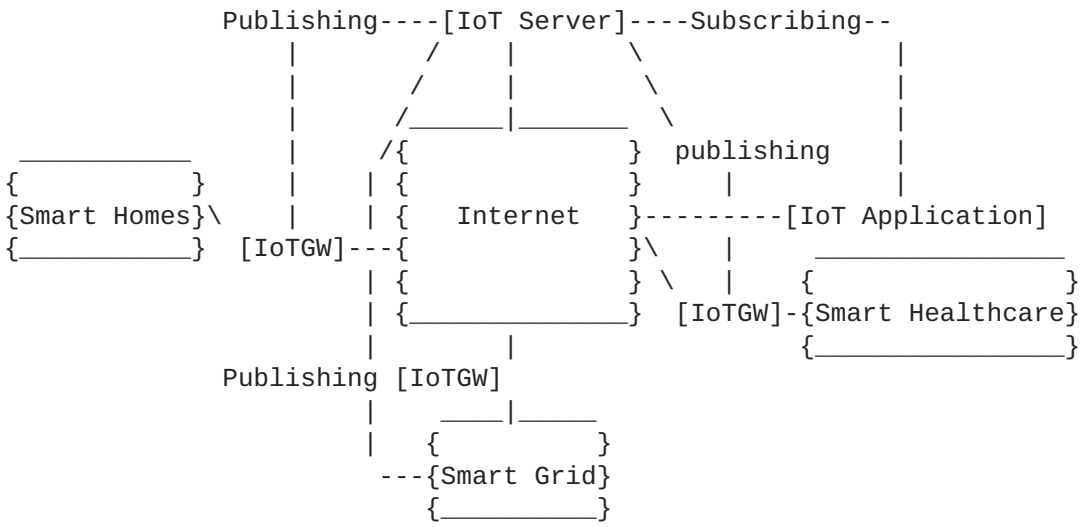

Figure 2: Implementing an open IoT architecture through standardized APIs on the IoT gateways and the server

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 we outlined in Section 3:

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

- Security and Trust. In IP, the security and trust model is based on session established between two hosts. Session-based protocols

rely on the exchange of several messages before a secure session is established. Use of such protocols in constrained IoT devices

can have serious consequences in terms of energy efficiency because transmission and reception of messages is often more costly than the cryptographic operations. The problem may be amplified with the number of nodes the constrained device has to interact with because of both the computation cost and per session key state required to be managed by the constrained device. Also the trust management schemes are still relatively weak, focusing on securing communication channels rather than managing the data that needs to be secured directly. Though key management in ICN is no less complex than in host based interactions, the benefits is associated with the security credentials in the content instead of the host. Trust is via keys that are bound to names through certificates whose private keys are held by the principals of the system, with IP focusing on the channel model of security while ICN focusing on the object model.

- 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. Further the 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, gateway or to the IoT server. Resource constraints of the IoT devices, especially in power and bandwidth, could seriously limit the performance of this approach. On the other hand, ICN supports in-network computing/caching/storage, which can alleviate this problem.

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

- Contextual Communications. This application-layer based IoT approach may not react to dynamic contextual changes in a timely fashion. The main reason is that context lists are usually kept at the IoT server in this approach, and they cannot help efficiently route requests information 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 enabled via name-based routing or name-based lookup.

- Self-Organization. The application-layer approach is topology-based as it is bound to IP semantics, and thus does not sufficiently satisfy the self-organization requirement. In addition to topological self-organization, IoT also requires data- and service-level self-organization [97], which is not supported by this approach.

- Ad-hoc and infrastructure mode. As mentioned above, the overlay-based approach lacks self-organization, and thus does not provide efficient support for the ad-hoc mode of communication.

4.2.2. Suitability of Delay Tolerant Networking(DTN)

In [21][22], delay-tolerant networking (DTN) has been considered to support future IoT architecture. DTN was created to support information delivery in the presence of network disruptions and disconnections, which has 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 (endpoint IDs), which usually correspond to a host or a service, and is mainly a way to transport data, while ICN provides a different paradigm centered around named data that addresses additional issues for IoT applications [23] through features such as information naming, information discovery, information request and dissemination. Also, the endpoint IDs could be used to also 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 transport would however still require other components from an ICN architecture such as naming conventions, so since the exact transport is not a major focus of the issues in this draft, most of of the discussions are applicable to a generic ICN architecture in general.

5. Advantages of using ICN for IoT

A key concept of ICN is the ability to name data independently from the current location at which it is stored, which simplifies caching and enables decoupling of sender and receiver. 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 general benefits that ICN could provide to IoT networks.

- Naming of Devices, Data and Services. The heterogeneity of both network equipment deployed and services offered by IoT networks 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 to the application layer the task to name data and services. This causes different applications to use different naming schemes, and no consistent mapping from application layer names to network names exist. In many common applications of IoT networks, data and services are the main goal, and ICN provides an intuitive way to name those in a way that can be utilized on the network layer as well. Communication with a specific device is often secondary, but when needed, the same ICN naming mechanisms can be used. The network distributes content and provides a service, instead of only sending data between two named devices. In this context, data content and services can be provided by several devices, or 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.

- Security and privacy. ICN advocates the model of object security to secure data in the network. This concept is based on the idea of securing information objects unlike 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 keys established at the application level. All of this means that the actual transmission of data does not have to be secured as the same security mechanisms protect the data after generation until consumed by a client, regardless of 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 could potentially be retrieved from a cache in a different location. Having a trust relationship with each of these different caches is not realistic.
  Through Name-Data Integrity, ICN automatically guarantees data integrity to the requester 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 in the device constraints are severe enough that it is not able to perform the required cryptographic operations for Object Security, 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; cryptographic hash of a public key also enables them to be self-certifying, i.e.,

authenticating the resource object does not require an external authority [25][26].

- 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 systems, because of their resource constraints. Wireless bandwidth and power supply can be limited for multiple devices sharing a communication channel, and for small mobile devices powered by batteries. In this case, avoiding unnecessary transmissions with IoT devices to retrieve and distribute IoT data to multiple places is important, hence processing and storing such content in the network can save wireless bandwidth and battery power. Moreover, as for other types of networks, applications for IoT networks requiring shorter delays can benefit from local caches and services to reduce delays between content request and delivery.

- Decoupling between Sender and Receiver. IoT devices may be mobile and face intermittent network connectivity. When 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.

6. 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 that will be involved in order to support large scale IoT deployment with diverse application requirements.

Though ICN integrates content/service/host abstraction, name-based routing, compute, caching/storage as part of the network infrastructure, IoT requires special considerations given heterogeneity of devices and interfaces such as for constrained networking [61][119][121], data processing, and content distribution models to meet specific application requirements which we identify as challenges in this section.

6.1. Naming Devices, Data, and Services

The ICN approach of named data and services (i.e., device independent naming) is typically desirable when retrieving IoT data. However, data centric naming may also pose challenges.

- Naming of devices: Naming devices can be useful in an IoT system.

For example, actuators require clients to act on a specific node of the deployed network, e.g. to switch it on or off; or it could be necessary to access to a particular device for administrator

purposes. This can be achieved through the specific name that uniquely identify the network entity of interest. Moreover, a persistent name allows a device to change attachment point without loosing its identity. A friendly way to address devices is using contextual hierarchical names, where the same types of names as for data objects can be used. To ensure that the device is always reached, it is important that it is possible to disable caching and request aggregation, if used, for such names.

- Size of data/service name: Content name can have variable length. 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), its length can reach high values. In particular, according to the specific application, content name size can exceed Data size. This can be the case of IoT sensed values that usually consist in few bytes: data could be as small as a short integer in case of temperature values, or one-byte in case of control messages of an actuator state (on/off). Moreover, a too long name would probably incur in fragmentation at the link layer, and related problems such as, several transmissions, delay and security issues. Viable solutions to handle ICN packets fragmentation and reassembly have been investigated in literature. For instance, the work in [105] proposes to perform the operations hop-by-hop: 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, thus 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 content is the one requested. This is only possible in contexts where the requested object is already existing, and where there is a directory service to look up names or learned through a manifest service. This approach is suitable for systems with large data objects where it is important to verify the content.

- Hierarchical names: The use of hierarchical names, such as in the CCN and NDN architectures make it easier to create names a priori and also provides a convenient way to use the same naming scheme for node names. Since the names are not self-certifying, this will require other mechanisms for verification of object integrity. If routing is also done on the hierarchical names, the

system will loose some of its location independence and caching will mostly only be done on the path to the publisher.

- Semantic and Metadata based content name: A semantic-based naming approach can allow a successful name retrieving through keywords

(for example, 'noise level at position X'), even if a perfect matching of name is not available [62]. Moreover, enriching contents with metadata allows to better describe them and to establish association between similar ones. However this mechanism require more advanced functionality for matching of such metadata in data objects to the semantics of the name (such as comparing the position information of an object with the position information of the requested name). The need for such potentially computationally heavy tasks in intermediate nodes in the network may be considered understanding the trade-offs in terms of application and network performance.

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

- Trust: Names can be used to verify the authenticity and integrity of the data. To provide security functionalities through names, it is possible to use different approaches. On one hand, hierarchical, schematized, Web-of-Trust models allow the public key verification. On the other hand, self-certifying names allow in-network integrity check of the name-key or name-content binding without the need of a Public Key Infrastructure (PKI) or other third party to establish whether the key is trustworthy or not. This can be realized (i) directly: the hash of the content is bound to the name; or (ii) indirectly: first, the hash of the content is signed with the secret key of the publisher, then the public key of the publisher and the signed hash are bound to the name [44]. The hash algorithm can be applied to already existing contents and where there is a directory service or manifests to look up names. In case of contents not yet published, but generated on demand, the hash cannot be known a priori. Thus, different trust mechanisms should be investigated. Moreover, self-certified names approach can hide content semantics, thus making names less human friendly. Since trends show that users prefer to find contents through search engine using keywords, non-human-friendly names could be a barrier unless the content is enriched with keywords. But, this problem does not concern M2M applications. In fact, human-readable names may not be useful in a context of just communicating machines.

- Flexibility: Further challenges arise for hierarchical naming schema: referring to requirements on "constructible names" and "on-demand publishing" [35][36]. TThe 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 also using partially specified names. The latter refers the possibility to request a content that has not yet been published in the past, thus triggering its creation.

- Scoping : From an application's point of view, scopes are used to gather related data. From the network's perspective, instead, scopes are used to mark where the content is available[65]. For instance, nodes involved in caching coordination can vary according to scope[66]. As a consequence, scoping allows to limit packet request propagation, improving bandwidth and energy resources usage, and control content dissemination thanks to access control rules, different for each scope[64]. However, 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 to limit interest propagation, provide a simple means to attain context-sensitive communication, etc. Finally, perimeter- and channel-based access control is often violated in 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 or more importantly violate privacy, 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 ICN naming scheme such as Attribute-Based Encryption [63] and access control delegation scheme [64]. In the first solution, a Trusted Third Party assigns a set of attributes to each network entity. Then, a publisher (i) encrypts the data with a random key; (II) generates the metadata for the decryption phase; (iii) creates an access policy used to encrypt the random key; (iv) appended 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.

6.2. 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 [57] [69] [70] [91]. 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 [50] [52]:

- 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 [57]. 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 [86] 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 [95]. In addition,

fast name lookup are necessary to ensure soft/hard real time services [106][107][108]. This challenge is especially important

for applications with stringent latency requirements, such as health monitoring, emergency handling and smart transportation [109].

- 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 [97].

- 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.

6.3. Security and Privacy

Security and privacy is crucial to all the IoT applications 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 [74], while at the same time the 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 TLS/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 [83], 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 data and

services are the main goal and specific communication between two devices is secondary. 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 [80] 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.

- In the resource-constrained nodes, energy limitation is the biggest challenge. Moreover, 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 [75] [55].

- In the infrastructure part, we have several new threats introduced by ICN-IoT [85] particularly in architectures employing name resolution service [119]. 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 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.

- As far as the IoT application server is concerned, data privacy is one of the biggest concerns. 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.

6.4. Caching

In-network caching helps bring data closer to consumers, but its usage differs in constrained and infrastructure part of the IoT network.

Caching in ICN-IoT faces several challenges:

- An important challenge is to determine which nodes on the routing path should cache the data. According to [52], caching the data on a subset of nodes can achieve a better gain than caching 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 [53], selecting a random router to cache data is as good as caching the content everywhere. In [88], the authors suggest that edge caching

provides most of the benefits of in-network caching typically discussed in NDN, with simpler deployment. However, it and other

papers consider workloads that are analogous to today's CDNs, not the IoT applications considered here. Further work is likely required to understand the appropriate caching approach for IoT applications.

- Another challenge in ICN-IoT caching is what to cache for IoT applications. In many IoT applications, customers often access a stream of sensor data, and as a result, caching a particular sensor data item for longer time may not be beneficial. In [90], proposed a caching scheme that ensures that older instances of the same sensor stream were first to be evicted from the cache when needed. In [55], the authors suggest to cache IoT services on intermediate routers, and in [57], the authors suggest to cache control information such as pub/sub lists on intermediate nodes. In addition, it is yet unclear 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 repeated actuation requests within a given time period), or have little meaning at all if actuation uses authenticated requests as in [89].

- Another challenge is that the efficiency of distributed caching may be application dependent. When content popularity is heterogeneous, some content is often requested repeatedly. In that case, the network can benefit from caching. Another case where caching would be beneficial is when devices with low duty cycle are present in the network and when access to the cloud infrastructure is limited. In [90], it is also shown that there are benefits to caching in the network when edge links are lossy, in particular if losses occur close to the content producer, as is common in wireless IoT networks. 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 request and later. It may also be less beneficial to have caches distributed throughout ICN nodes 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 variable content/state. For example, when the last value for a sensor reading is requested or desired, the returned data should change every time the sensor reading is updated. In that case, ICN caching may increase the risk that cache inconsistencies result in old data being returned.

6.5. Storage

Storage is useful for IoT systems both at longer and small time scales.

Long terms storage can be distributed at vantage points including both the edge and the main IoT service aggregation points such as in the data centers, the difference being in the size of data, processing intelligence and heterogeneity of information that has to be dealt at the two points. The purpose of long terms storage at the edge is to analyze, filter, aggregate and re-publish data for consumption by either by the parent service components or directly by the consumers. The aggregation service points, republish data to be presented as part of the global pub/sub service to interested consuming parties. Long term storage for IoT data also serves the purpose of data backup and replication. Specifically, we face several issues here. Firstly, we need to decide how many replicas we should have for each stream of IoT data, and where we should store these replicas. Given that many IoT applications consume data locally, storage locations should be kept near to data sources as well. Since IoT data are mostly appended to the end of a stream, instead of being updated, managing multiple replicas becomes easier. Secondly, we need to adopt a mechanism that can efficiently route traffic to the nearest data replica. ICN provides several solutions to this problem. For example, global name resolution service (GNRS) can keep track of each replica's location [56].

Short-term in-network storage (here storage refers to temporary buffer when an outgoing link is not available) helps improve communication reliability, especially when network links are unreliable, such as wireless links. ICN-IoT could adopt a generalized storage-aware routing algorithm to support delay and disruption tolerance in the routing layer. Each router employs in-network storage that facilitates store vs. forward decisions in response to varying link quality and disconnections [111]. These decisions are based on both short-term and long-term path quality metrics. In addition, packets along paths that become disconnected are handled by a disruption tolerant networking (DTN) mode of the protocol with delayed delivery and replication features. In particular, each router maintains two types of topology information: (i) An intra-partition graph is formed by collecting flooded link state advertisements which carry fine-grained, time-sensitive information about the intra-network links; (ii) A DTN graph is maintained via epidemically disseminated link-state advertisements which carry connection probabilities between all nodes in the network. In-network storage faces the following challenges: (1) when to store and how long to store the data, and (2) the next step after the short-term storage. In [90] the authors also shows 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.

6.6. Routing and Forwarding

ICN-IoT supports both device-to-device (D2D) communication and device-to-infrastructure (D2I) communication. Some D2D communications are within a single IoT domain, while others might cross IoT domains involving 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 sensor network popular routing protocols such as RPL [81], AODV[82], etc. The main challenge it faces is the resource constraint of the IoT nodes. In order to address this challenge, we could adopt a light-weight, much shorter ICN name for each communicating party within an IoT domain (see Section 6.12 for details). Before we leave the IoT domain, the gateway node will translate the party's short ICN name to its original ICN name. Data forwarding in the ICN infrastructure part can adopt either direct name-based routing or indirect routing using a name resolution service (NRS).

- In direct name-based routing, packets are forwarded by the name of the data [91][61][71] or the name of the destination node [72]. Here, the main challenge is to keep the ICN router state required to route/forward data low. This challenge becomes more serious when a flat naming scheme is used due to the lack of aggregation capabilities.

- In indirect routing, packets are forwarded based upon the locater of the destination node, and the locater is obtained through the name resolution service. In particular, the name-locater binding can be done either before routing (i.e., static binding) or during routing (i.e., dynamic binding). For static binding, the router state is the same as that in traditional routers, and the main challenge is the need to have fast name resolution, especially when the IoT nodes are mobile. For dynamic binding, ICN routers need to main a name-based routing table, hence the challenge of keeping the state information low. At the same time, the need of fast name resolution is also critical.

6.7. Mobility Management

Considering the diversity of IoT applications mobility ranges from tracking sensor data from mobile human beings to large fleets of diverse mobile elements such as drones, vehicles, trucks, trains associated with a transport infrastructure. These mobility could be over heterogeneous access infrastructure ranging from short range 802.15.4 to cellular radios. Further, handling information delivery in ad hoc setting involving vehicles, road side units (RSU) and the corresponding infrastructure based services offers more challenges. ICN architectures has generally been shown to handle consumer and producer mobility [59], and even suitability to V2V scenarios [60]. Networking tools to handle mobility varies with application requirements, which varies from being tolerant to packet losses and latency to those that are mission critical with stringent requirement on both these QoS metrics.

Related to this, the challenge is to quantify the cost associated with mobility management both in the control and forwarding plane, to handle both static binding versus dynamic binding (dynamic binding here refers to enabling seamless mobility) of named resources to its location when either or both consumer and producer is mobile.

During a network transaction, either the data producer or the consumer may move away and thus we need to handle the mobility to avoid information loss. ICN may differentiate mobility of a data consumer from that of a producer:

- When a consumer moves to a new location after sending out the request for Data, the Data may traverse to the previous point of attachment (PoA) but leaving copies of it through its previous path, which can be retrieved by the consumer by retransmitting its request, a technique used by direct routing approach. Indirect routing approach doesn't differentiate between consumer and producer mobility [91], as it only requires an update to the name resolution system, which can update the routers to rebind the named resource to its new location, while using late-binding to route the packet from the previous PoA to the new one.

- If the data producer itself has moved, the challenge is to control the control overhead while searching for a new data producer (or for the same data producer in its new position) [58]. To this end, flooding techniques could be used rediscover the producer, or the direct routing techniques can be enhanced with late-binding feature to enable seamless mobility [59].

6.8. Contextual Communication

Contextualization through metadata in ICN control or application payload allows IoT applications to adapt to different environments. This enables intelligent networks which are self-configurable and enable intelligent networking among consumers and producers [55]. For example, let us look at the following smart transportation scenario: "James walks on NYC streets and wants to find an empty cab closest to his location." In this example, the context is the relative locations of James and 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, naming conventions could be used to 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 in the 100m grid area 4QFJ 123 678 of 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 is very challenging:

- We need to have a fast context resolution service through which the involved IoT devices can continuously update its contextual information to the application (e.g., each taxi's location and Jame's information in the above example). Or, in the namespace driven approach, mechanisms for continuous nearest neighbor queries in the namespace need to be developed.

- The difficulty of this challenge grows rapidly when the number of devices involved in a context as well as the number of contexts increases.

6.9. 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, in which each sensor node or ICN router implements context reasoning [55]. Another major purpose of in-network computing is to filter and cleanse sensed data in IoT applications, that is critical as the data is noisy as is [73].

Named Function Networking [113] describes an extension of the ICN concept to named functions processed in the network, which could be used to generate data flow processing applications well-suited to, for example, time series data processing in IoT sensing

applications.

Related to this, is the need to support efficient function naming. Functions, input parameters, and the output result could be encapsulated in the packet header, the packet body, or mixture of the two (e.g. [31]). If functions are encapsulated in packet headers, the naming scheme affects how a computation task is routed in the network, which IoT devices are involved in the computation task (e.g. [54]), and how a name is decomposed into smaller computation tasks and deployed in the network for a better performance.

Another is challenge is related to support computing-aware routing. Normal routing is for forwarding requests to the nearest source or cache and return the data to the requester, whereas the routing for in-network computation has a different purpose. If the computation task is for aggregating sensed data, the routing strategy is to route the data to achieve a better aggregation performance [51].

In-network computing also includes synchronization challenges. Some computation tasks may need synchronizations between sub-tasks or IoT devices, e.g. a device may not send data as soon as it is available because waiting for data from the neighbours may lead to a better aggregation result; some devices may choose to sleep to save energy while waiting for the results from the neighbours; while aggregating the computation results along the path, the intermediate IoT devices may need to choose the results generated within a certain time window.

6.10. Self-Orgnization

General IoT deployments involves heterogeneous IoT systems consisting of embedded systems, aggregators and service gateways in a IoT domain. To scale IoT deployments to large scale, scope-based self-organization is required. This relates to IoT system middleware functions [118] which include device bootstrapping and discovery, assigning local/global names to device and/or content, security and trust management functions towards device authentication and data privacy. ICN based on-boarding protocols have been studied [96] and has shown to offer significant savings compared to existing approaches. These challenges span both the constrained devices as well as interaction with the aggregators and the service gateways which may have to contact external services like authentication servers to on-board devices. A critical performance optimization metric of these functions while operating at scale is to have low control and data overhead in order to maximize energy efficiency. Further, in the infrastructure part 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.

6.11. 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 leveraging state-based forwarding in protocols like CCN/NDN. However these features have not been analyzed from the QoS perspective when heterogeneous traffic patterns are mixed in a router, in general QoS for ICN is an open area of research [121]. In-network reliability comes at the cost of a complex network layer; hence the research challenges here is to build redundancy and reliability in the network layer to handle a wide range of disruption scenarios such as congestion, short or long term disconnection, or last mile wireless impairments. Also an ICN network should allow features such as opportunistic store and forward mechanism to be enabled only at certain points in the network, as these mechanisms also entail overheads in the control and forwarding plane overhead which will adversely affect application throughput, Please see the discussion on in-network storage (Section 6.5) for more details .

6.12. Resource Constraints and Heterogeneity

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

global reach-ability and mobility handling yet still considers resource constraints of embedded devices.

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

7. Differences from T2TRG

T2TRG [9] is a IoT research group under IRTF focusing on research challenges of realizing IoT solutions considering IP as the narrow waist. IP-IoT has been a research topic over a decade and with active industry solutions, hence this group provides an venue to study advanced issues related to IP-IoT security, provisioning, configuration and inter-operability considering various heterogeneous application environments. ICN-IoT is a recent research effort, where the objective to exploit ICN feature of name based routing and security, caching, multicasting, mobility etc in an end-to-end manner to enable IoT services spanning both ad hoc, infrastructure and hybrid scenarios. More detailed comparison of IP-IoT versus ICN-IoT is given in Section 4.

8. Security Considerations

ICN puts security in the forefront of its design which ICN-IoT can leverage to build applications with varying security requirements, which has been discussed quite elaborately in this draft. This is an informational draft and doesn't create new considerations beyond what has been discussed.

9. 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 towards 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 application and networking perspective. We then discuss why current approach of application layer unified IoT solutions based on IP falls short of meeting these requirements, and how ICN architecture is a more suitable towards this. 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 varying IoT service needs.

10. Acknowledgements

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

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

[20] 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.

[21] 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.

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

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

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

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

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

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

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

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

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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 .

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

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

[47] 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.

[48] 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.

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

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

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

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

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

[54] 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.

[55] 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.

[56] 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.

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

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

[59] 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.

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

[61] 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.

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

[63] 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..

[64] 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..

[65] 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..

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

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

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

[69] 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.

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

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

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

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

[74] 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.

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

[76] 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.

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

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

[79] 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, <http://www.rfc-editor.org/info/rfc7230>.

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

[81] 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, <http://www.rfc-editor.org/info/rfc6550>.

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

[83] marc.mosko@parc.com, m., Uzun, E., and C. Wood, "CCNx Key Exchange Protocol Version 1.0", draft-wood-icnrg-ccnxkeyexchange-01 (work in progress), October 2016.

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

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

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

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

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

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

[90] 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.

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

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

[93] 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.

[94] 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.

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

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

[97] 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.

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

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

[100] 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.

[101] 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..

[102] 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..

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

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

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

[106] 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.

[107] 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.

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

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

[110] 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.

[111] 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.

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

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

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

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

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

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

[118] 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.

[119] 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.

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

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

Authors' Addresses

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

Email: yyzhang@winlab.rutgers.edu

Prof. Dipankar Raychadhuri
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: ray@winlab.rutgers.edu

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

Email: alfredo.grieco@poliba.it

Prof. Emmanuel Baccelli
INRIA
Room 148, Takustrasse 9
Berlin 14195
France

Email: Emmanuel.Baccelli@inria.fr

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

Email: jburke@ucla.edu

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

Email: ravi.ravindran@huawei.com

Guoqiang Wang
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: gq.wang@huawei.com

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

Email: anders.lindgren@ri.se

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

Email: bengt.ahlgren@ri.se

Olov Schelen
Lulea University of Technology
Lulea SE-971 87
SE

Email: lov.schelen@ltu.se