ICN Research Group
Internet-Draft
Intended status: Informational
Expires: October 24, 2016

Y. Zhang
D. Raychadhuri
WINLAB, Rutgers University
L. Grieco
Politecnico di Bari (DEI)
E. Baccelli
INRIA
J. Burke
UCLA REMAP
R. Ravindran
G. Wang
Huawei Technologies
A. Lindren
B. Ahlgren
SICS Swedish ICT
O. Schelen
Lulea University of Technology

April 22, 2016

Requirements and Challenges for IoT over ICN
draft-zhang-icnrg-icniot-requirements-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 forming a horizontal unified, defragmented IoT platform. Such a platform will make objects accessible to applications across organizations and domains. Towards this goal, quite a few proposals have been made to build a unified host-centric IoT platform as an overlay on top of today’s host-centric Internet. However, there is a fundamental mismatch between the host-centric nature of todays Internet and the information-centric nature of the IoT system. To address this mismatch, we propose to build a common set of protocols and services, which form an IoT platform, based on the Information Centric Network (ICN) architecture, which we call ICN-IoT. ICN-IoT leverages the salient features of ICN, and thus provides seamless mobility support, security, scalability, and efficient content and service delivery.

This draft describes representative IoT requirements and ICN challenges to realize a unified ICN-IoT framework. Towards this, we first identify a list of important requirements which a unified IoT architecture should have to support tens of billions of objects, then we discuss how the current IP-IoT overlay fails to meet these requirements, followed by discussion on suitability of ICN for IoT.

Though we see most of the IoT requirements can be met by ICN, we discuss specific challenges ICN has to address to satisfy them. Then we provide discussion of popular IoT scenarios including the "smart" home, campus, grid, transportation infrastructure, healthcare, Education, and Entertainment for completeness, as specific scenarios requires appropriate design choices and architectural considerations towards developing an ICN-IoT solution.

Status of This Memo

Copyright Notice

Table of Contents

1. IoT Motivation

During the past decade, many standalone Internet of Things (IoT) systems have been developed and deployed in different domains. The recent trend, however, is to evolve towards a globally unified IoT platform, in which billions 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. Building a unified IoT platform, 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, 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 motes that will provide the asked information). Taking a general IoT perspective, we first discuss the IoT requirements generally applicable to many well known scenarios. We then discuss how the current IP overlay models 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 challenges from an ICN perspective and requirements posed towards its design. Final discussion focuses on IoT scenarios and their unique challenges.

2. 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.

2.1. Naming

The first step towards realizing a unified IoT platform is the ability to assign names that are unique within the scope and lifetime of each device, data items generated by these devices, or a group of devices towards a common objective. Naming has the following requirements: first, names need to be persistent (within one or more contexts) against dynamic features that are common in IoT systems, such as lifetime, mobility or migration; second, names need to be secure based on application requirements; third, names should provide advantages to application authors in comparison with traditional host address based schemes.

2.2. 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. In addition, 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.

2.3. Resource Constraints

IoT devices can be broadly classified into two groups: resource-sufficient and resource-constrained. 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. 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.

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 and jitter. This in particular applies to satellite or other space based devices.

2.4. Traffic Characteristics

IoT traffic can be broadly classified into local area traffic and wide area traffic. Local area traffic is between 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.

2.5. Contextual Communication

Many IoT applications shall rely on dynamic contexts in the IoT system to initiate communication between 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. 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.

2.6. Handling Mobility

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

Mobility in the IoT platform 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 and if necessary to negotiate different connectivity or security constraints specific to each mobile context.

2.7. 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. In a unified IoT platform, depending on application requirements, content caching may or may not be policy driven. If caching is pervasive, intermediate nodes don't need to always forward a content request to its original creator; rather, locating and receiving a cached copy is sufficient for IoT applications. This optimization can greatly reduce the content access latencies.

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

In network storage and caching, however, has the following requirements on the IoT platform. The platform needs to support the efficient resolution of cached copies. Further the platform should strive for the balance between caching, content security/privacy, and regulations.

2.8. Security and Privacy

In addition to the fundamental challenge of trust management, a variety of security and privacy concerns also exist in ICNs.

The unified IoT platform 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 [11].

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 platform. Privacy means that both the content and the context around IoT data need to be protected. These requirements will be driven by various stake holders such as industry, government, consumers etc.

2.9. 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. Reliable communication requires the following capabilities for the underlying system: (1) seamless mobility support in the face of extreme disruptions (DTN), (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 communication patterns (unlike the tree-like routing structure supported by RPL developed by ROLL WG).

2.10. Self-Organization

The unified IoT platform 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 platform-wide 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 platform 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 platform, 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 infrastructure [23] [24].

2.11. 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 platform needs to design a common protocol that serves both modes. Such a protocol should be able to provide: (1) energy-efficient topology discovery and data forwarding in the ad-hoc mode, and (2) scalable name resolution in the infrastructure mode.

2.12. Open API

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 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. Examples include pull, push, and publish/subscribe mechanisms using common naming, payload, encryption and signature schemes.

2.13. 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 [18][19][20]. 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, ephemeral nature 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.

3. 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 platform in which the existing silo IoT systems, as well as new systems that are rapidly deployed. This will make their data and services accessible to general Internet applications (as in

ETSI- M2M and oneM2M standards). In such a unified platform, resources can be accessed over Internet and shared across the physical boundaries of the enterprise. However, current approaches to achieve this objective are based upon Internet overlays, whose inherent inefficiencies due to IP protocol [8] hinders the platform from satisfying the IoT requirements outlined earlier (particularly in terms of scalability, security, mobility, and self-organization)

3.1. Silo IoT Architecture

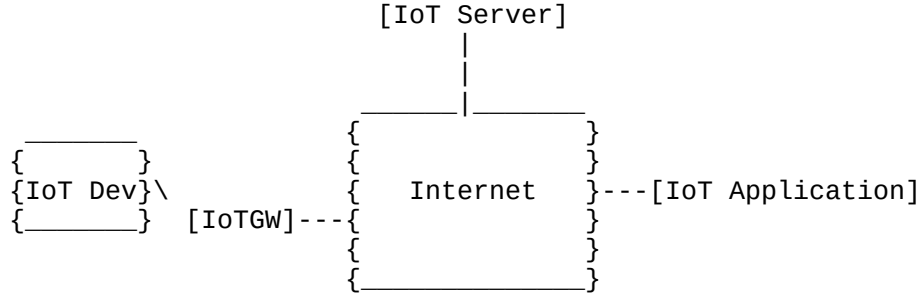

Figure 1:Silo architecture of standalone IoT systems

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, leading to a highly fragmented protocol space with limited interoperability.

3.2. Overlay Based Unified IoT Solutions

The current approach to a unified IoT platform 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 as an overlay, this is the most practical approach towards a unified IoT platform. 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) [49], that is a lightweight protocol modeled after HTTP [50] 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 platform) has both the protocol bindings to HTTP and CoAP for its primitives. Figure 2 shows the architecture adopted in this approach.

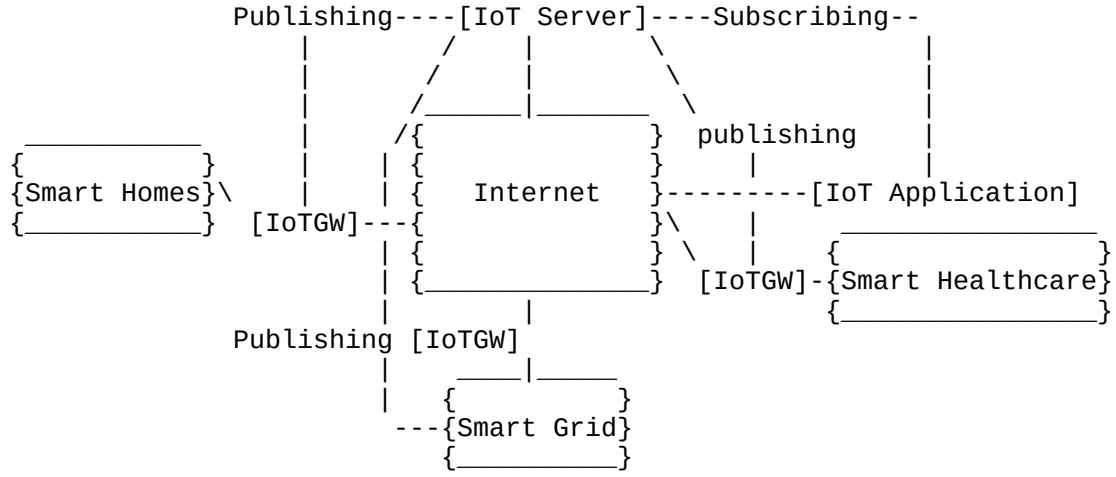

Figure 2: Implementing an open IoT platform through standarized APIs on the IoT gateways and the server

3.2.1. Weaknesses of the Overlay-based Approach

The above overlay-based 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 2:

- Naming. In current overlays for IoT systems the naming scheme is host centric, i.e., the name of a given resource/service is linked

to the one of 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 provided by many different devices thus requiring a different design rationale.

- Trust. Trust management schemes are still relatively weak, focusing on securing communication channels rather than managing the data that needs to be secured directly.

- Mobility. The overlay-based 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 overlay-based approach requires every device to send data to an aggregator or to the IoT server. Resource constraints of the IoT devices, especially in power and bandwidth, could seriously limit the performance of this approach.

- Traffic Characteristics. In this approach, applications are written in a host-centric manner suitable for point-to-point communication. IoT requires multicast support that is challenging in overlay systems today.

- Contextual Communications. This overlay-based approach cannot react to dynamic contextual changes in a timely fashion. The main reason is that context lists are kept at the IoT server in this approach, and they cannot help efficiently route requests information at the network layer.

- Storage and Caching. The overlay-based 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 overlay-based 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 [59], which is not supported by the overlay 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.

4. 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 such advantages compared to using traditional host-centric networks. 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. In many common applications of IoT networks, data and services are the main goal, and specific communication between two devices is secondary. The network distributes content and provides a service, instead of establishing a communication link between two 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.

- 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.

5. ICN Challenges for IoT

This section outlines some of the ICN specific challenges [71] that must be considered when defining an IoT framework over ICN, and describes some of the trade offs that will be involved.

ICN integrates content/service/host abstraction, name-based routing, compute, caching/storage as part of the network infrastructure connecting consumers and services which meets most of the requirements discussed above; however IoT requires special considerations given heterogeneity of devices and interfaces such as for constrained networking [38][70], data processing, and content distribution models to meet specific application requirements which we identify as challenges in this section.

5.1. Naming Devices, Data, and Services

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 is often important in an IoT network. The presence of actuators requires clients to act specifically on a device, e.g. to switch it on or off. Also, managing and monitoring the devices for administration purposes requires devices to have a specific name allowing to identify them uniquely. There are multiple ways to achieve device naming, even in systems that are data centric by nature. For example, in systems that are addressable or searchable based on metadata or sensor content, the device identifier can be included as a special kind of metadata or sensor reading.

- Size of data/service name: In information centric applications, the size of the data is typically larger than its name. For the IoT, sensors and actuators are very common, and they can generate or use data as small as a short integer containing a temperature value, or a one-byte instruction to switch off an actuator. The name of the content for each of these pieces of data has to uniquely identify the content. For this reason, many existing naming schemes have long names that are likely to be longer than the actual data content for many types of IoT applications. Furthermore, naming schemes that have self certifying properties (e.g., by creating the name based on a hash of the content), suffer from the problem that the object can only be requested when the object has been created and the content is already known, thus requiring some form of indexing service. While this is an acceptable overhead for larger data objects, it is infeasible for use when the object size is on the order of a few bytes.

- 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. This approach is suitable for systems with large data objects where it is important to verify the content.

- Metadata-based content name: Relying on metadata allows to generate a name for an object before it is created. However this mechanism requires metadata matching semantics.

- Naming of services: Similarly 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: We need to ensure the name of a network element is issued by a trustworthy issuer in the context of the application, such as a trusted organization in [44]. Further the validity of each piece of data published by an authorized entity in the namespace should be verifiable - e.g., by following a hierarchical chain-of-trust to a root that is acceptable for the application. See [54]s for an example.

- Flexibility: Further challenges arise for hierarchical naming schema: referring to requirements on "constructible names" and "on-demand publishing" [28][29]. The former entails that each user is able to construct the name of a desired data item through specific algorithms and that it is possible to retrieve information 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.

- Control/scoping : Some information could be accessible only within a given scope. This challenge is very relevant for smart home and health monitoring applications, where privacy issues play a key role and the local scope of a home or healthcare environment may be well-defined. However, 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, mechanisms for name confidentiality should be available in the ICN-IoT architecture.

5.2. Name Resolution

Inter-connecting numerous IoT entities, as well as establishing reachability to them, requires a scalable name resolution system considering several dynamic factors like mobility of end points, service replication, in-network caching, failure or migration [37] [40] [41] [57]. 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 [31] [33]:

- 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 [37]. 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 [53] may offer further scalability and efficiency.

- Deployability and interoperability: 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 interoperate.

- 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 [58]. In addition, fast name lookup are necessary to ensure soft/hard real time

services [60][61][62]. This challenge is especially important for applications with stringent latency requirements, such as health monitoring, emergency handling and smart transportation [63].

- 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 [59].

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

5.3. Caching/Storage

In-network caching helps bring data closer to consumers, but its usage differs in constrained and infrastructure part of the IoT network. Caching in constrained networks is limited to small amounts in the order of 10KB, while caching in infrastructure part of the network can allow much larger chunks.

Caching in ICN-IoT faces several challenges:

- The main challenge is to determine which nodes on the routing path should cache the data. According to [33], 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 [34], selecting a random router to cache data is as good as caching the content everywhere. In [55], 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 may not be beneficial. In [36], the authors suggest to cache IoT services on intermediate routers, and in [37], 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 [56].

- 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. 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.

5.4. Routing and Forwarding

Routing in ICN-IoT differs from routing in traditional IP networks in that ICN routing is based upon names instead of locators. Broadly speaking, ICN routing can be categorized into the following two categories: direct name-based routing and indirect routing using a name resolution service (NRS).

- In direct name-based routing, packets are forwarded by the name of the data [57][38][42] or the name of the destination node [43]. 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 locator of the destination node, and the locator is obtained through the name resolution service. In particular, the name-locator 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. Finally, another challenge is to quantify the cost associated with mobility management, especially static binding vs. dynamic binding.

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 get lost, which requires the consumer to simply resend the request, a technique used by direct routing approach. Indirect routing approach doesn’t differentiate between consumer and producer mobility [57], also network caching can improve data recovery for this approach.

- 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). To this end, flooding techniques could be used, but an intra-domain level only, otherwise the network stability would be seriously impaired. For handling mobility across different domains, more sophisticated approaches could be used, including the adoption of a SDN-based control plane.

5.5. 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 [36]. 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.

5.6. 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 [36]. 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 [44].

Named Function Networking [64] 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. [24]). 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. [35]), 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 [32].

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.

5.7. Security and Privacy

Security and privacy is crucial to all the IoT applications applications including the use cases discussed in Section 5. In one recent demonstration,it was shown that passive tire pressure sensors in cars could be hacked and used as a gateway into the automotive system [38]. 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 trust in network hosts. 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. This reinforces an inherent characteristic of ICN networks i.e. to decouple senders and receivers. 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. 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.

Implementing security and privacy methods faces different challenges in the constrained and infrastructure part of the network.

- In the resource-constrained nodes, energy limitation is the biggest challenge. As an example, let us look at a typical sensor tag. Suppose the tag has a single 16-bit processor, often running at 6 MHz to save energy, with 512Bytes of RAM and 16KB of flash for program storage. Moreover, it has to deliver its data over a wireless link for at least 10,000 hours on a coin cell battery. As a result, traditional security/privacy measures are impossible to be implemented in the constrained part. In this case, one possible solution might be utilizing the physical wireless signals as security measures [46] [36].

- In the infrastructure part, we have several new threats introduced by ICN-IoT [52]:

  1. We need to ensure the name of a network element is issued by a trustworthy organization entity such as in [48], or by its trusted delegate. As name securely binds to data in ICN, security constraints of content that has not yet been published yet should also be taken into consideration.

  2. An intruder may gain access or gather information from a resource it is not entitled to. As a consequence, an adversary may examine, remove or even modify confidential information.

  3. An intruder may mimic an authorized user or network process. As a result, the intruder may forge signatures, or impersonate a source address.

  4. An adversary may manipulate the message exchange process between network entities. Such manipulation may involve replay, rerouting, mis-routing and deletion of messages.

  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.

5.8. Self-Orgnization

General IoT deployments involves heterogeneous IoT systems or subsystems within a particular scenario. Here scope-based self-organization is required to ensure logical isolation between the IoT subsystems, which should be enabled at different levels -- device/service discovery, naming, topology construction, routing over logical ICN topologies, and caching [69]. These challenges are

extended to constrained devices as well and they should be energy and device capability aware. In the infrastructure part, intelligent name-based routing, caching, in-network computing techniques should be studied to meet the scope-based self-configuration needs of ICN-IoT.

5.9. 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 [71]. 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.

5.10. Energy Efficiency

All the optimizations for other components of the ICN-IoT system (described in earlier subsections) can lead to optimized energy efficiency. As a result, we refer the readers to read sections 5.1-5.9 for challenges associated with energy efficiency for ICN-IoT.

6. Appendix

Several types of IoT applications exists, where the goal is efficient and secure management and communication among objects in the system and with the physical world through sensors, RFIDs and other devices. Below we list a few popular IoT applications. We omit the often used term "smart", though it applies to each IoT scenario below, and posit that IoT-style interconnection of devices to make these environments "smart" in today’s terms will simply be the future norm.

6.1. Homes

The home [10] is a complex ecosystem of IoT devices and applications including climate control, home security monitoring, smoke detection, electrical metering, health/wellness, and entertainment systems. In a unified IoT platform, we would inter-connect these systems through the Internet, such that they can interact with each other and make

decisions at an aggregated level. Also, the systems can be accessed and manipulated remotely. Challenges in the home include topology independent service discovery, common protocol for heterogeneous device/application/service interaction, policy based routing/ forwarding, service mobility as well as privacy protection. Notably, the ease-of-use expectations and training of both users and installers also presents challenges in user interface and user experience design that are impacted by the complexity of network configuration, brittleness to change, configuration of trust management, etc. Finally, it is unlikely that there will be a single "home system", but rather a collection of moderately inter-operable collaborating devices. In addition, several IoT-enabled homes could form a smart district where it becomes possible to bargain resources and trade with utility suppliers.

Homes [12][13] faces the following challenges that are hard to address with IP-based overlay solutions: (1) context-aware control: home systems must make decisions (e.g., on how to control, when to collect data, where to carry out computation, when to interact with end-users, etc.) based upon the contextual information [14]; (2) inter-operability: home systems must operate with devices that adopt heterogeneous naming, trust, communication, and control systems; (3) mobility: home systems must deal with mobility caused by the movement of sensors or data receivers; (4) security: a home systems must be able to deal with foreign devices, handle a variety of user permissions (occupants of various types, guests, device manufacturers, installers and integrators, utility and infrastructure providers) and involve users in important security decisions without overwhelming them; (5) user interface / user experience: homes need to provide reasonable interfaces to their highly heterogeneous IoT networks for users with a variety of skill levels, backgrounds, cultures, interests, etc.

Smart homes have the following specific requirement for the underlying architecture:

- Smart homes require names that can enable local and wide area interactions; Also, security, privacy, and access control is particularly important for smart homes.

- Smart homes may use in-network caching at gateway to enable efficient content access.

- In smart homes, we need local, intra-domain and inter-domain routing protocols.

- In smart homes many control loops and actions depend heavily on the context, and the contexts evolve with time, e.g., temperature, weather, number of occupants, etc.

- In smart homes, local services can provide value-added contributions to a standardized home gateway network, through features such as reporting, context-based control, coordination with mobile devices, etc.

- In smart homes, the access to networked information should be shielded to protect the privacy of people, for example, cross-correlation of device activity patterns to infer higher-level activity information.

6.2. Enterprise

Enterprise building deployments, from university campuses [15] [65] [66] [67] to industrial facilities and retail complexes, drive an additional set of scalability, security, and integration requirements beyond the home, while requiring much of its ease of use and flexibility. Additionally, they bring requirements for integration with business IT systems, though often with the additional support of in-house engineering support.

Increasing number of enterprises are equipped with sensing and communication devices inside buildings, laboratories, and plants, at stadiums, in parking lots, on school buses, etc. A unified IoT platform must integrate many aspects of human interaction, H2M and M2M communication, within the enterprise, and thus enable many IoT applications that can benefit a large body of enterprise affiliates. The challenges in smart enterprise include efficient and secure device/data/resource discovery, inter-operability between different control systems, throughput scaling with number of devices, and unreliable communication due to mobility and telepresence.

Enterprises face the following challenges that are hard to address with IP-based overlay solutions: (1) efficient device/data/ resource discovery: enterprise devices must be able to quickly and securely discover requested device, data, or resources; (2) scalability: a enterprise system must be able to scale efficiently with the number and type of sensors and devices across not only a single building but multi-national corporations (for example); (3) mobility: a enterprise system must be able to deal with mobility caused by movement of devices; (4) security: security for IoT applications in the enterprise should integrate with other enterprise-wide security components.

6.3. Smart Grid

Central to the so-called "smart grid"[16] is data flow and information management, achieved by using sensors and actuators, which enables important capabilities such as substation and distribution automation. In a unified IoT platform, data collected from different smart grids can be integrated to achieve more optimizations that include reliability, real-time control, secure communications, and data privacy.

Deployment of the smart grid [17] [21] faces the following issues that are hard to address with IP-based overlay solutions: (1) scalability: future electrical grids must be able to scale gracefully to manage a large number of heterogeneous devices; (2) real time: grids must be able to perform real-time data collection, data processing and control; (3) reliability: grids must be resilient to hardware/software/networking failures; (4) security: grids and associated systems are often considered critical infrastructure -- they must be able to defend against malicious attacks, detect intrusion, and route around disruption.

Smart grids have the following specific requirements for the underlying IoT architecture:

- Smart grids require names and name resolution system that can enable networked control loops, real-time control, and security.
- Smart grids may use in-network caching to back up valuable data improving reliability.
- In smart grids, we often require very timely data delivery. Therefore, it is important to be able to locate the closest information. In addition, routing/forwarding robustness and resilience is also critical.
- In smart grids, contextual information such as location, time, voltage fluctuations, depending on the specific segment of the grid, can be used to optimize several power distribution objectives.
- In smart grids, we often rely on in-network computing to increase the scalability and efficiency of the system, putting computation closer to the data sources.
- In smart grids, energy consumptions profiles should never be disclosed at a fine granularity as it can be used to violate user privacy.

6.4. Transportation

We are currently witnessing the increasing integration of sensors into cars, other vehicles transportation systems [22]. Current production cars already carry many sensors ranging from rain gauges and accelerometers over wheel rotation/traction sensors, to cameras. These sensors can not only be used for internal vehicle functions, but they could also be networked and leveraged for applications such as monitoring external traffic/road conditions. Further, we can build vehicle-to- infrastructure (V2I),Vehicle-to-Roadside (V2R), and vehicle-to- vehicle (V2V) communications that enable many more applications for safety, convenience, entertainment, etc. The challenges for transportation include fast data/device/service discovery and association, efficient communications with mobility, trustworthy data collection and exchange.

Transportation [22][25] faces the following challenges that are hard to address with IP-based overlay solutions: (1) mobility: a transportation system must deal with a large number of mobile nodes interacting through a combination of infrastructure and ad hoc communication methods; ; also, during the journey the user might cross several realms, each one implementing different stacks (whether ICN or IP); (2) real-time and reliability: transportation systems must be able to operate in real-time and remain resilient in the presence of failures; (3) in-network computing/filtering: transportation systems will benefit from in-network computing/ filtering as such operations can reduce the end-to-end latency; (4) inter-operatibility: transportation systems must operate with heterogeneous device and protocols; (5) security: transportation systems must be resilient to malicious physical and cyber attacks.

Smart transportation applications have the following specific requirements for the underlying IoT architecture:

- Smart transportation systems require names and name resolution system to be able to handle extreme mobility, short latency and security. In addition, the mobility patterns of transportation systems increase the likelyhood that a user migrates from one network realm to another one during the journey. In this case, names and NRS should be designed in such a way to enable interoperability between different heterogeneous ICN realms and/or ICN and IP realms [68].

- Smart transportation may implement in-network caching on vehicles for efficient information dissemination

- In smart transportation, vehicle-to-vehicle ad-hoc communication is required for efficient information dissemination.

- In smart transportation, many different contexts exist, intertwined to each other and highly changing, which include location - both geographical and jurisdictional, time - absolute and relative to a schedule, traffic, speed, etc.

- In smart transportation, in-network computing is very useful to make vehicle become an active element of the system and to improve response time and scalability.

- In smart transportation, the habits of users can be inferred by looking at their movement patterns -- privacy protection is essential.

6.5. Healthcare

As more embedded medical devices, or devices that can monitor human health become increasingly deployed, healthcare is becoming a viable alternative to traditional healthcare solutions [26]. Further, consumer applications for managing and interacting with health data are a burgeoning area of research and commercial applications. For future health applications, a unified IoT platform is critical for improved patient care and consumer health support by sharing data across systems, enabling timely actuations, and lowering the time to innovation by simplifying interaction across devices from many manufacturers. Challenges in healthcare include real-time interactions, high reliability, short communication latencies, trustworthy, security and privacy, and well as defining and meeting the regulatory requirements that should impact new devices and their interconnection. In addition to this dimension, assistive robotics applications are gaining momentum to provide 24/24 7/7 assistance to patients [59].

Healthcare [26][27] faces the following challenges that are hard to address with IP-based overlay solutions: (1) real-time and reliability: healthcare systems must be able to operate on real-time and remain resilient in the presence of failures; (2) inter-operability: healthcare systems must operate with heterogeneous devices and protocols; (3) security: healthcare systems must be resilient to malicious physical and cyber attacks and meet the regulatory requirement for data security and interoperability; (4) privacy: user trust in healthcare systems is critical, and privacy considerations paramount to garner adoption and continued user; (5) user interface / user experience: the highly heterogeneous nature of real-world healthcare systems, which will continue to increase through the introduction of IoT devices, presents significant challenges in interface design that may have architectural implications.

Smart healthcare applications have the following specific requirements for the underlying IoT architecture:

- Smart healthcare system requires names and name resolution system to enable real- time interactions, dependability, and security.
- Smart healthcare may use in-network caching for rapid information dissemination.
- In smart healthcare, timely and dependable routing and information forwarding is the key.
- In smart healthcare several contexts can be used to delineate between levels of care and urgency, for example delineating between chronic, everyday, urgent, and emergency situations. Such contexts can evolve rapidly with significant impact to individuals health. Hence timely and accurate detection of contexts is critical.
- In smart healthcare, in-network computing can help resolve contexts and ensure security and dependability, as well as provide low-latency responses to urgent situations.
- In smart healthcare, personal medical data about patients should remain shielded to protect their privacy, implementing both regulatory requirements and current industry best practices.

6.6. Education

IoT technologies enable the instrumentation of a variety of environments (from greenhouses to industrial plants, homes and vehicles) to support not only their everyday operation but an understanding of how they operate -- a fundamental contribution to education. The diverse uses of hobbyist-oriented micro-controller platforms (e.g., the Arduino) and embedded systems (e.g., the Raspberry PI) point to a burgeoning community that should be supported by the next generation IoT platform because of its fundamental importance to formal and informal education.

Educational uses of IoT deployments include both learning about the operation of the system itself as well as the systems being observed and controlled. Such deployments face the following challenges that are hard to address with IP-based overlay solutions: (1) relatively simple communications patterns are obscured by many layers of translation from the host-based addressing of IP (and layer 2 configuration below) to the name-oriented interfaces provided by developers; (2) security considerations with overlay deployments and channel-based limit access to systems where read-only use of data is

not a security risk; (3) real-time communication helps make the relationship between physical phenomena and network messages easier to understand in many simple cases; (4) integration of devices from a variety of sources and manufacturers is currently quite difficult because of varying standards for basic communication, and limits experimentation; (5) programming interfaces must be carefully developed to expose important concepts clearly and in light of current best practices in education.

Smart campus systems have the following specific requirements for the underlying IoT architecture:

- Smart campus systems usually consist of heterogeneous IoT services, thus requiring names and name resolution system to enable resource/ service ownership, and be application-centric.
- Smart campus systems may use in-network caching to enable social interactions and efficient content access.
- In smart campus, inter-domain routing protocols are required which often need short latency.
- In smart campus, due to the existence of many services, relevant contextual inputs can be used to improve the quality and efficiency of different services.
- In smart campus, in-network computing services can be used to provide context for different applications.
- In smart campus, it is required to differentiate among different profiles and to allocate different rights and protection levels to them.

6.7. Entertainment, arts, and culture

IoT technologies can contribute uniquely to both the worldwide entertainment market and the fundamental human activity of creating and sharing art and culture. By supporting new types of human-computer interaction, IoT can enable new gaming, film/video, and other "content" experiences, integrating them with, for example, the lighting control of the smart home, presentation systems of the smart enterprise, or even the incentive mechanisms of smart healthcare systems (to, say, encourage and measure physical activity).

Entertainment, arts, and culture applications generate a variety of challenges for IoT: (1) notably, the ability to securely "repurpose" deployed smart systems (e.g., lighting) to create experiences; (2) low-latency communication to enable end-user responsiveness; (3)

integration with infrastructure-based sensing (e.g., computer vision) to create comprehensive interactive environments or to provide user identity information; (4) time synchronization with audio/video playback and rendering in 3D systems (5) simplicity of development and experimentation, to enable the cost- and time-efficient integration of IoT into experiences being designed without expert engineers of IoT systems; (6) security, because of integration with personal devices and smart environments, as well as billing systems.

7. Informative References

[1] Cisco System Inc., CISCO., "Cisco visual networking index: Global mobile data traffic forecast update.", 2009-2014.

[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://www.nets-fia.net/", 2010.

[9] 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.

[10] 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.

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

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

[13] 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.

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

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

[16] 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.

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

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

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

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

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

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

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

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

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

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

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

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

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

[35] 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.

[36] 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.

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

[38] 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.

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

[40] 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.

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

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

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

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

[45] 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.

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

[47] 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.

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

[49] 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>.

[50] 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>.

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

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

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

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

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

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

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

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

[59] 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", Computer Communications, Volume 54, 1 December, 2014.

[60] 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.

[61] 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.

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

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

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

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

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

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

[68] Project, BonVoyage., "From Bilbao to Oslo, intermodal mobility solutions, interfaces and applications for people and goods, supported by an innovative communication network", Call H2020-MG-2014, 2015-2018.

[69] 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.

[70] 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.

[71] 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

Andres Lindgren
SICS Swedish ICT
Box 1263
Kista SE-164 29
SE

Email: andersl@sics.se

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

Email: bengta@sics.se

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

Email: lov.schelen@ltu.se