ISI: Integrate Sensor Networks to Internet with ICN

Sripriya S. Adhatarao, Mayutan Arumaithurai, Dirk Kutscher and Xiaoming Fu

Abstract—Internet of Things (IoT) is a growing topic of interest. Billions of IoT devices are expected to connect to the Internet in the near future. These devices differ from the traditional devices operated in the Internet. In this work, we argue that an Information Centric Networking (ICN), a new networking paradigm, is a more suitable architecture for the IoT compared to the currently prevailing IP based network. We observe that recent works that propose to use ICN for IoT, either do not cover the need to integrate *Sensor Networks* with the Internet to realize IoT or do so inefficiently. There is a need to understand effective ways to integrate the various heterogeneous *Sensor Networks* with the Internet without affecting their current mode of operation. In this work, we study the essential requirements for integrating *Sensor Networks* to the Internet. We provide an architecture with *Gateways* for paving a way for the *Sensor Networks* to become a part of the IoT family. We further provide a naming schema for efficient operation of the resource constrained *Sensor Networks*, discuss mobility, security, communication patterns and propose the most suitable choices for IoT networks.

Index Terms— Sensor Networks, Internet, IoT, ICN, Gateway.

1 Introduction

Internet of Things (IoT) refers to a network of devices like machines, vehicles, electronic appliances, and also wearables like Radio Frequency IDentification tags (RFID), step-counter, etc. These devices are usually embedded with sensors, actuators, memory and network connectivity. They are mainly used for sensing, monitoring and controlling various applications. IoT is a growing topic of interest and has already drawn attention of academia and Industry.

A current popular topic of interest in the IoT community is *smart cities*. The smart cities are in design phase and will come to reality in the near future. The IoT devices will be deployed extensively in the infrastructures like smart buildings, smart offices, etc. Additionally, IoT's have found applications in many different fields like smart homes, smart vehicles, industries, environmental monitoring, etc.

Billions of IoT devices are expected to be connected in the near future to communicate, sense and gather data. However, it is important to understand the difference between IoT and *Sensor Networks*. There are various heterogenous *Sensor Networks* that operate in their own private network. The goal of IoT is to provide the *Sensor Networks* with access to Internet in order for them to become IoT [1].

Currently, IoTs are designed to operate with the IP architecture [2]. However, IoT networks often contain many resource constrained devices with smaller memory, limited computational capacity and power supply (mostly a battery). Many IoT applications require devices to operate for longer periods in remote locations with no facilities *e.g.* forests. Due to constraints, IoT devices are equipped with Layer2 technologies like IEEE 802.15.4 and Bluetooth LE; hence, they operate with a much smaller MTU than the current MTU used in the Internet. They also incur several other challenges like limited IP address space, while point-to-point connectivity is heavy for these resource constrained devices. Additionally security is another critical aspect in many IoT applications and is expensive (induces overhead) to achieve with IP leading to complexity in operation and resource consumption. There is therefore a need for an efficient design for the IoT devices that is scalable, efficient and provides a secure mechanism for communication to gather data for monitoring and/or controlling the devices. Shang et al. [3] discuss many of such issues in detail.

We observed that IoTs are usually studied/researched as separate entities (see §10). However, that should not be the case. One has to also consider all the potential issues in the Internet. Isolating them might lead to unforseen consequences. IoT needs well connected networks to communicate and pass information/control messages to other devices in the network. One might also question, what is the benefit of integrating the *Sensor Networks* with the Internet? We believe that by integrating the *Sensor Networks* with the Internet, both the networks can benefit. Primarily, *Sensor Networks* will benefit from the existing features of the Internet thus incorporating the ubiquitous *Sensor Networks* into the IoT world. At the same time, we see a scope for the Internet to widen.

Information-Centric Networking (ICN) [4], [5] is a new networking paradigm. In ICN, content is treated as the first-class entity and nodes exchange information based on the *Names* of the content instead of the IP addresses of the end points that request or provide the information. This shift from a "location-based" network to a "content-centric" network entails efficiency for content dissemination, especially when the content may be available at multiple points and also when the provider and/or consumer are mobile. Since many ICN designs incorporate extensive (in-network)

- *S.Adhatarao is a Ph.D student in Institut für Informatik, University of Göttingen. Email: adhatarao@cs.uni-goettingen.de*
- *Dr. M. Arumaithurai is a senior researcher in Institut für Informatik, University of Göttingen. Email: arumaithurai@cs.uni-goettingen.de*
- *Dr. D. Kutscher is the CTO for Virtual Networking and IP at Huawei's German Research Center in Munich. Email: dku@dkutscher.net.*
- *Prof.Dr. X. Fu is the Head of the Computer Networks Group, Institut für Informatik, University of Göttingen. Email: fu@cs.uni-goettingen.de*

caching, additional performance benefits can be realized with those widespread caches. ICN is growing rapidly with a highly active community. Researchers are increasingly proposing new and improved solutions in various areas like routing, forwarding, caching, naming, etc.

Similar to ICN, in IoT the devices are interested in the content and not their location *i.e.*, IoT's are information centric in nature. The design of ICN suits fairly well for IoTs. There is no longer a need for maintaining point-point communication. Recent ICN proposals such as Named Data Networking (NDN [4]) and Content Oriented Publish/Subscribe System (COPSS [5]), which enhances NDN with an efficient publish/subscribe capability, adopt human-readable, hierarchically structured Names and Content Descriptors (CDs). The namespace is unbounded and can easily support billions of devices and even more. There is an increased availability of data due to in-network caching. Security is enhanced since ICN embeds security in the content rather than on the communication link. The multicast and broadcast features are easily supported with ICN without additional overhead. Although, there have been many instances where NDN has been explored for the Internet and in some cases for supporting the IoT environments [3], [6], [7] we argue that, IoT does not need the full Content Centric Networking (CCN)/NDN stack. Some other works like CCN-lite/NDN-lite [8] support more specifically the resource constrained IoT devices.

The Internet can represent either the existing TCP/IP, ICN or both the architectures. In this work, we mainly focus on the Internet running ICN protocols because the various Internet Engineering Task Force (IETF) groups (*e.g.* 6LoWPAN [9], Core, ROLL) [10] are already focusing on integrating the IoT devices to the IP based Internet. With ICN in the Internet, we see an increased amount of caching in the Internet. Hence the IoT networks will incur less traffic. On the other hand, the Internet users can have access to IoT data without having to explicitly follow a different protocol for the multitude of IoT networks. The users can also control the IoT devices via Internet. This can also simplify the application designs.

Many recent works that advocate the use of ICN for IoT focus on aspects such as data retrieval patterns [11], routing [12], benefits of caching [13] and architectural changes [14]. The Internet houses powerful devices that are capable of running the full fledged ICN protocols like NDN. However, the *Sensor Networks* do not need the entire ICN stack and would require lighter versions of the ICN protocols. Hence like recent works we assume that the *Sensor Networks* will operate with lighter versions of the ICN protocols.

Even though the recent works like [12], [14], etc. have focused on solving many different problems of IoT, they have not considered the crucial aspect of the need for an architecture to integrate the *SensorNetworks* running lighter version of ICN protocol with the Internet in order to pave a way for them to join the IoT family. Therefore in this work, we address this vital need. The main goal of this work is to connect the multitude of *Sensor Networks* to the Internet, i.e., to connect these two worlds together. We look at the various requirements for realizing such an ICN based – IoT=Internet+SN architecture. We propose to introduce *Gateways* to integrate the *Sensor Networks* with the Internet.

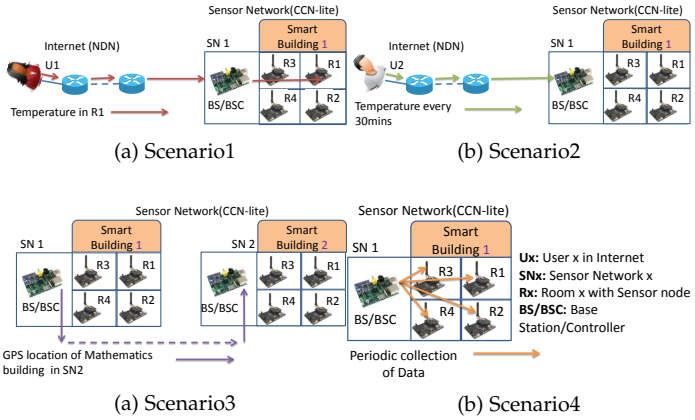

Fig. 2: Use Cases.

We identify the explicit functions/responsibilities of such a *Gateway* and the various services offered by it. We design a naming schema that can efficiently support the *Sensor Networks* running ICN protocols. Additionally, we also analyze the various communication patterns, mobility and security for IoT devices.

The contributions in this paper include:

- Analysis of the requirements for an architecture to integrate *Sensor Networks* with the Internet
- Architecture *ISI*: to integrate *Sensor Networks* to the Internet with ICN
- Naming schema for the IoT networks running ICN
- Communication protocol for the IoT networks
- Discussion on the aspect of Mobility and Security for IoT

2 Use Case Formalization

In this section we formalize some of the essential use cases that will derive the requirements for integrating *Sensor Networks* to the Internet and will further assist in building the *ISI* architecture.

We take the example of a smart city as shown in Figure.2 with many buildings equipped with *Sensor Networks* to build our use cases. Let us consider an application of monitoring the temperature of all the rooms in a smart building as a representative *Sensor Network*. The smart building has several rooms equipped with temperature sensitive machinery *e.g.* servers and hence each room is equipped with a temperature sensing sensor device. There is a Base Station (BS) that gathers the temperature from each room every 30mins and raises an alarm when abnormalities occur. Several such scenarios can be gathered in a smart city *e.g.* in Industrial units, offices, smart houses for controlling fire, etc. Let's say, the *Sensor Network1* (SN1) is operating in the building1 which is a Computer Science building in a university, and *Sensor Network2* (SN2) is operating in the building2 which is a Mathematics building in the same university.

Scenario 1: A user (U1) (*e.g.* a system administrator) in the Internet is interested in the temperature of room1 in the building1. U1 is in Internet that runs an ICN protocol like NDN [15]. However the building1 is located in SN1 that runs a lighter version of an ICN protocol like CCN-lite [8]. Besides, U1 is interested in the temperature of room1 sensed by the sensor device S1 which is not awake all the time in order to save its constrained resources like power. U1 is not aware of the time at which S1 will be available to serve the requested content, nor the exact protocol that it runs.

Scenario 2: A user (U2) in the Internet is interested in receiving the cumulative temperature of the building1 every 30mins. Similar to scenario1, U2 is in Internet but would like to receive the content collected and computed by the BS located in the *Sensor Network*.

Scenario 3: The Base Station1 (BS1) in SN1 needs some content *e.g.* the GPS location of the Mathematics building in the smart city. The mathematics building is located in SN2 of the smart city. The SN2 might be running the same or a different ICN protocol from SN1. BS1 is un aware of the location of the content or the protocol used in SN2.

Scenario 4: Let us consider the operation of SN1. The sensor devices sense the temperature in each room periodically every 30mins. The BS1 is interested in the content produced by these sensing devices. The BS1 will gather the content and compute the cumulative temperature of the entire building and will raise an alarm when it observes abnormalities. What would be the most efficient way to collect this information that can ensure maximum utilization of the available resources in the constrained devices in SN1?

With the help of the above mentioned use cases we show the need for a protocol translation between Internet and *Sensor Network* and among *Sensor Networks* running different protocols, the lack of information availability about the nature of the sensor network by the users in the Internet, the different kinds of contents requested by the users from Internet and other *Sensor Network* and the issue of efficient utilization of the resources to gather information within a *Sensor Network*. Although we use these four use cases here to derive the requirements for *ISI*, we can derive many more requirements and even different requirements based on the nature of the *Sensor Network* used as a representative. However, these basic use cases can be applied to all *Sensor Networks* and *ISI* can easily support additional use cases.

3 Requirement Analysis

In this section we build on our use cases to derive the various requirements for integrating the *Sensor Networks* with the Internet. We assume that *Sensor Networks* are operating with a lighter version of the ICN protocol like CCN-lite (or NDN-lite) while the Internet operates with the full ICN stack like complete NDN stack. We choose these architectures as representative for ICN, similar requirements will apply to other ICN architectures like MobilityFirst [16], etc.

Gateway: We observe from use case 1, 2 and 3 that there are multitude of users scattered across hybrid *Sensor Networks* and Internet. However these networks don't use the same protocol. There is a need for protocol translation among these networks for communicating with each other. An efficient way to interface networks operating different protocols is via a *Gateway*. The *Gateway* should run both the protocols of the networks it serves. The *Gateway* should be equipped with necessary intelligence and data structures to perform near transparent flow of traffic between the two networks to seamlessly integrate these networks.

Naming: Since ICN is a name based protocol, we observe from all the four use cases a need for a naming schema for the IoT devices that is more catered towards the operation of the constrained devices in the *Sensor Networks*. The rationale for this requirement is the MTU in the *Sensor Networks* is much smaller (127B) when compared to the MTU (1280B) used by the devices in the Internet. ICN supports, unbonded and any length namespace. There is a need for smaller ICN names that can not only fit into the MTU of the *Sensor Networks* devices but also requires less storage on the forwarding engines of the sensing devices. This will ensure that IoT networks resources are used efficiently and will also ease their scalability.

Communication Protocol: All the use cases present a different form of communication either query/response (1&3) or pub/sub (2&4). In many *Sensor Networks*, there is a Base Station (BS) that collects the data sensed by the sensors in the network. Another common pattern is a Base Station Controller (BSC) controls the sensor devices by sending control messages to them. We can notice that, unlike IP, the users in these networks are interested in the content similar to ICN irrespective of their location. With an efficient communication protocol we can ensure efficient utilization of the constrained resources in the *Sensor Networks*. E.g. in use case 2&4, the user U2 and BS can use pub/sub to collect the data periodically. Additionally, with ICN in Internet we can benefit with the caching as it not only reduces the traffic in the core network, but will also reduce the traffic entering the resource constrained IoT networks. This ensures maximum availability of the content even when the sensing devices are sleeping to save resources.

Mobility: In addition to the identified usecases, another and a more important aspect to consider is the mobility of sensor devices. The IoT networks should also be able to handle the mobility of sensor devices from one domain to another. Mobility may affect the naming, reachability and other aspects of the moving devices.

Security: Security is a greater concern in many if not all of the *Sensor Networks* and Internet. There has been a growing concern that IoT's are designed without addressing many of the associated security concerns [17]. However, security induces additional overhead especially in the *Sensor Networks*. There is a need to analyze and provide some security measures that meet the security requirement of the *Sensor Networks* and are not an overkill with regards to their constrained resources.

4 NDN Architecture

To proceed further it is important to briefly discuss the ICN architectures we use as representative in this work.

The CCN forwarding engine model contains three main components: the Content Store (CS), the Pending Interest Table (PIT) and the Forwarding Information Base (FIB).

The NDN communication protocol usually begins with a user interested in some content generating an Interest with the respective ContentName. When the Interest arrives at a CCN router in the forwarding engine of the router the CS is checked to see if the content with the same name already exists in which case the content is returned. In case of a CS miss PIT is checked to see if an Interest with the same name has already been forwarded. If yes, then the incoming face of the Interest is recorded. If not, an entry is added to the PIT and the FIB is checked to find the forwarding face and the Interest is forwarded. When a Data packet arrives, the CS is checked, and if a matching entry is present then the Data packet is discarded otherwise the CS stores the Data and PIT is checked. If an entry is found in the PIT then the

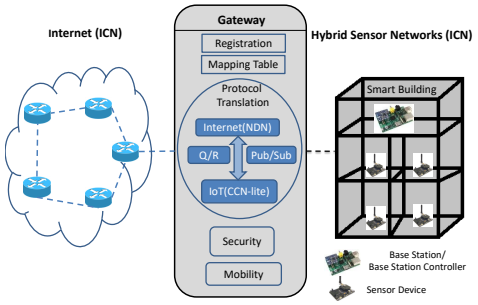

Fig. 3: The *ISI* Architecture.

Data is forwarded to all the outgoing faces recorded in the PIT for this ContentName. Due to space constraint, we don't discuss the COPSS architecture and point readers to [5] for further reading.

5 Architecture

In this section we describe our *ISI* architecture shown in Figure. 3 for integrating the various hybrid *Sensor Networks* with the Internet running ICN protocols.

5.1 Components

NDN network: We choose NDN as a representative ICN architecture. In principal this could be any of the ICN architectures. The network is capable of retrieving *Content* from publishers in both the Internet as well as the IoT network. The network is also capable of delivering control messages to the *Sensor Networks*.

Sensor Networks: There are many possible *Sensor Networks* that represent numerous applications. The *Sensor Networks* usually contain hybrid devices. The devices can have wireless or wired access. They can be operating for the purpose of monitoring, controlling, etc. As discussed earlier, in this work we propose the *Sensor Networks* to operate with the ICN protocols. We will use a lighter version of the CCN called CCN-lite [8] for the *Sensor Networks*. The *Sensor Networks* usually contain a *Base Station (BS)* or *Base Station Controller (BSC)* that collect the data sensed by other devices in the network for monitoring purpose or to control their operation. The *BS/BSC* are usually powerful machines and not constrained in resources.

Gateway: This is the most essential piece of the proposed design. The *Gateway* sits between the Internet running the NDN protocol and the *Sensor Networks* running CCN-lite protocol. All of the traffic to/from the Internet and *Sensor Networks* has to pass through the *Gateway*. The main function of *Gateway* is to perform protocol translation between the two networks. The *Gateway* is also responsible for mapping functions (discussed shortly). It is clear that since the IPV6 MTU is 1280B while the IEEE 802.15.4 can support only 127B, the authors in the IETF standard [2] suggest that header compression in IPV6 is unavoidable. However, the content generated in the *Sensor Network* is assumed to be small. We believe that with efficient designs of ICN names in the *Sensor Network* the *Gateway* does not have to compress the headers. Although we speak about one *Gateway* between the Internet and each *Sensor Network*, there is no restriction on the number of *Gateways*. As the traffic exchange between the Internet and *Sensor Networks* increases, the burden on a single *Gateway* also increases. Hence, multiple *Gateways* have to be used to distribute the load.

5.2 Description

As said earlier we assume that the Internet is running the NDN protocol. There are many different *Sensor Networks* that represent various applications like environmental monitoring, smart houses, etc. that are running the CCN-lite protocol. The users are spread across both the Internet as well as the *Sensor Networks*. The aim of this design is to let the applications operate in the *Sensor Networks* as they desire but extend their availability and control by integrating them with the Internet. The design allows the users in Internet to access/control the IoT devices using the Internet. A key to achieve this integration is through *Gateways*.

The *Gateway* is a powerful component that has many roles to play. Every *Sensor Network* is associated with one or more *Gateways* that is responsible for seamlessly integrating the respective *Sensor Network* with the Internet. The entire traffic between these two worlds will flow through the *Gateway* transparent to the users in both the networks.

The *Gateway* runs both the NDN protocol and CCN-lite protocol. The *Gateway* maintains a mapping table which maps the lengthy, unbounded names form the Internet running the NDN protocol to their equivalent short names in the *Sensor Networks* running the CCN-lite protocol.

The *Gateway* uses a registration procedure for every device in the *Sensor Network*. Each device upon entering the network must register itself with the *Gateway*. The *Gateway* provides an ID to each newly added device. The device then registers the short name and long name of the Content that it wishes to serve. These entries will be added to the mapping table maintained in the *Gateway*. The mapping table should be updated whenever there are any changes in the Content served by the *Sensor Network*.

We will use the term *Inbound traffic* for the traffic entering the *Sensor Network* and the term *Outbound traffic* for the traffic going out of *Sensor Network*. The inbound traffic from the users can be either a request for data or a request containing a control message. There are two possible outbound traffic. One containing the reply to the inbound traffic and the other is the request/subscription traffic generated inside the *Sensor Network*. The two types of outbound traffic should be distinguished form one another as the reply traffic needs a name change through a mapping table lookup. This can be achieved by using any one bit available field in the packets.

When the *Gateway* receives inbound traffic it is basically a NDN Interest packet. The *Gateway* scans the mapping table to find the equivalent short name. The *Gateway* creates a CCN-lite Interest packet with the short name and forwards it to the *Sensor Network*. Upon receiving a CCN-lite Data packet from the *Sensor Network*, the *Gateway* performs a lookup in the mapping table to find the long name and creates a NDN Data packet with the long name, extracts the Content from the CCN-lite Data packet and inserts it into the NDN Data packet and forwards it in the Internet.

Another type of out bound traffic is basically a CCN-lite Interest for content located either in the Internet or in other IoT network. The *Gateways* also support inter IoT network communication. When an IoT network running for a certain application needs information from outside its network, it simply generates a CCN-lite Interest and forwards it to the *Gateway*. Since this is an out bound request traffic the *Gateway* simply translates it to a NDN Interest and forwards it

in the network. It eventually reaches the intended publisher in Internet or the *Gateway* associated with the target *Sensor Network*. When the publisher is located in Internet, it follows the standard NDN protocol and reply with the data packet. If the Interest reaches a *Gateway* associated to another *Sensor Network*, the *Gateway* performs a mapping table lookup to fetch the equivalent short name and prepares a CCN-lite Interest and forwards it to the *Sensor Network*. Upon receiving the Data packet it prepares the NDN Data packet as explained earlier and forwards it to the Internet. The data packet finally reaches the *Gateway* of the *Sensor Network* that initiated the request. The *Gateway* performs a mere protocol translation and generates the CCN-lite Data packet as described earlier and forwards it to the *Sensor Network*.

6 Naming Schema

In this section we discuss the Naming schema for the IoT.

6.1 Naming in IoT

The IoT devices usually come with the Ethernet technology like IEEE 802.15.4 and Bluetooth LE. This results in a much smaller MTU (127B) compared to the traditional layer-2 technologies adopted in the Internet. This raises a concern on the size of the packets that traverse the IoT network. One way to reduce the packet size is to have smaller names that are relevant to the IoT networks. Since the any length, unbounded hierarchically structured names defined by many ICN architectures do not suit the IoT networks. The names should be precise to serve the purpose of the application and yet be specific and small.

We describe our naming schema using the same example of the smart city with *Sensor Networks* as shown in Figure. 3. Let us consider the same application of monitoring the temperature of all the rooms in a smart building from our use cases in §2. We know each room in the smart building is equipped with temperature sensing sensor devices and BS collects the data generated by the sensing devices.

We propose a naming structure of the form *Metric/ID/Area/Date/Time*. The first component *Metric* specifies what kind of data is generated by the sensing device E.g. temperature, pressure, humidity, etc. The second component *ID* indicates an identifier assigned to the device. The third component *Area* specifies the location/geographic range covered by the sensing device E.g. room, building, GPS location, etc. The fourth component *Date* specifies the day at which the readings are measured and the fifth component *Time* specifies the time at which the reading was captured by the device. The granularity of each component can be application dependant E.g. the time can vary from hours to seconds to minutes or more. Applying this naming schema to our smart building example the temperature sensed by a device with the id 01 in the room1 on 3rd November 2016 at 12:30 could be retrieved with a name */temp/01/r1/03-11-16/12:30*. This name is only 26B, leaving the rest of the packet for the content.

In most *Sensor Networks* the BS usually collects the data periodically from the sensing devices in the network. To distinguish the data retrieved from the BS we can use a naming schema of the form */Metric/BS/Area/Date/Time*. Note that the component *Area* in the naming schema for IoT represents individual rooms whereas it represent the whole building in case of the BS. So a user in NDN network can request for temperature of the whole building or for the individual rooms. This naming schema is fairly general and similar names can be created for different IoT networks based on specifics of the area or application of the IoT devices used in the network.

6.2 Naming in Internet

Recent ICN proposals such as Named Data Networking (NDN [4]) and Content Oriented Publish/Subscribe System (COPSS [5]) adopt human-readable, hierarchically structured Names and Content Descriptors (CDs).

Continuing our smart building example, a possible name structure for renaming the content for the Internet could be */temperature/UNI/ComputerScience/Building1/03-11-16/12:30* for the temperature of the Computer Science building in the university on 3rd November 2016 at 12:30 and */temperature/UNI/ComputerScience/Building1/room1/03-11-16/12:30* for the temperature of room1 at 12:30. We allow the availability of data sensed in every room in Internet as some rooms might be sensitive to temperature and would have to be monitored E.g. the temperature of a server room.

7 Communication Protocols

In this section we discuss the current communication protocols and propose the suitable communication protocol for IoT and CCN network.

7.1 Query/Response (Q/R) Communication

This is a dominant mode of communication in current networks. A user interested in some content simply queries the network. Any producer of the data (or also a node with an available copy of the data in case of ICN) responds to the request.

7.1.1 Q/R Communication in IoT

In *Sensor Network* the BS can query the sensing devices to retrieve the sensed data. The BS simply generates a request with the respective ContentName and forwards it to the network. Upon reaching the producer, the sensing device responds with the requested content. The BS can also send control messages using the query/response mode where the query can contain the control command while the response can contain an acknowledgement of the action taken. The IoT devices can also query for content located in Internet or other IoT networks. The *Gateway* should assist in retrieving the content in this case.

7.1.2 Q/R Communication in Internet

A user in Internet generates a ICN request with the ContentName of the desired content and forwards it in to the network. If the content is located in the caches of any intermediate forwarding node then the cached copy is returned to the user. Otherwise the Interest eventually reaches the producer who replies with the requested content. If the producer is located in a IoT network, then the *Gateway* will assist in retrieving the content however, this will be transparent to the user.

7.2 Publish/Subscribe (pub/sub) Communication

In a pub/sub scenario, there are two roles to play: publisher and subscriber. The publishers usually generate some data that could be of interest to subscribers. The subscribers

maintain a long term subscriptions to the content published by the publishers (refer to COPSS [5] for detail). Whenever a piece of content is published by the publishers, the network will deliver it to all of its subscribers.

7.2.1 Pub/sub Communication in IoT

In IoT the sensing devices that periodically sense the data can take up the role of publisher and publish the sensed data periodically. The BS is the subscriber interested in these contents and will subscribe to these contents and hence will receive them when published.

7.2.2 Pub/sub Communication in Internet

Similar to IoT, publishers and subscribers exist in ICN too. The users in ICN can not only subscribe to the content published inside the Internet but also by the IoT networks.

7.3 Communication protocol for IoT

There are two possible scopes of communication that can take place in IoTs: one is within the *Sensor Networks* and the other is with Internet/other Sensor Networks. The nature of communication in these network differs and can greatly affect the design choices of an efficient communication protocol for the *Sensor Networks*.

In *Sensor Networks*, the devices are usually sensing the respective data periodically and the BS gathers the sensed data and analyzes it to take appropriate actions. The Pub/Sub mode of communications seems desirable in this scenario. The devices can behave like publishers and publish the data periodically. While the BS can be a subscriber that subscribes to the data published by all the devices in its *Sensor Network*. This design choice allows the resource constrained sensor devices to save the battery by waking up only to sense and publish the data. We agree, that there are chances for the packet to get lost. So, for reliability reason we propose the BS to resume to Q/R to retrieve only the data that was lost during the next periodic cycle when the sensing device is awake to sense the next reading. However, during Q/R we expect the sensor device to wait for an acknowledgement before going back to sleep to ensure the BS has received the packet this time. This also means the sensing device should retain some of its previously sensed data, which we believe can be configured based on the applications requirement E.g. three most recent readings for reliability.

In the Internet, we see that Q/R is a dominant form of communication. However, for communicating with the Sensor Networks we can choose between Q/R and Pub/Sub based on the need of the user. If the user (or any other application or *Sensor Network*) is interested in periodically receiving all of the data collected by the *Sensor Networks*, then Pub/Sub looks like an ideal choice in this scenario. To reduce the burden on resource constrained sensing devices in the *Sensor Network*, we believe the caching can be enabled on BS. Other sensing devices can choose to retain or turn off caching based on their available resources. Since the BS gathers the data from all the Sensors, it can act as a publisher to the subscribers in the Internet. Please note that the subscribers in the Internet will subscribe to the longer names and not the shorter names used for publishing inside the *Sensor Networks*. So the *Gateway* has to create new publication data packet with their equivalent long names from the mapping table and then forward it to the users in Internet.

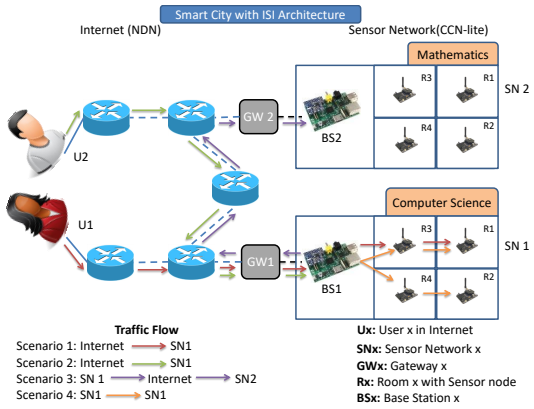

Fig. 4: Smart City Use Cases with ISI Architecture.

Another and most likely a common scenario is when users in the Internet are interested in some particular data generated in the *Sensor Networks* or would like to send some control message to the *Sensor Networks*. Intuitively, Q/R seems ideal for this scenario. When the user is interested in some data they just generate a CCN Interest with the longer Content Name and the network forwards it to the *Gateway*. The *Gateway* then performs a protocol translation and generates a CCN-lite Interest with the equivalent short name from the mapping table and forwards it to the BS/BSC. Based on the type of the request received the BS/BSC either replies with the requested data or an acknowledgement for the action taken.

8 Use Case Realization

In this section we discuss the functionality of each component of the proposed architecture in greater detail with the help of the use cases we defined in §2.

Consider the example network topology of a smart city with *ISI* architecture as shown in Figure. 4. There is a Base Station (BS) in each *Sensor Networks* that collects the temperature sensed by the devices periodically every 30mins. Each network is attached to a *Gateway* that runs both the NDN and CCN-lite protocols for interconnecting the *Sensor Networks* and the Internet.

Scenario 1: A user (U1) in the Internet is interested in the temperature of room1 in the building1. So, U1 will generate a NDN Interest with the name */temperature/UNI/ComputerScience/Building1/room1/03-11-16/12:30*. The network will forward the Interest to *Gateway*1 (GW1). GW1 will check its mapping table for the name. When no match is found, the GW1 behaves like a router and the NDN module running inside the GW1 will handle this packet by forwarding it to the appropriate router in the Internet. If there is a match, the GW1 will generate a CCN-lite Interest with the equivalent short name */temp/01/r1/03-11-16/12:30* and forward it to the CCN-lite module which will forward it to the Base Station1 (BS1). Since the BS1 periodically collects all the data the BS1 will generate the CCN-lite data with the requested content and forward it to GW1 otherwise it is forwarded to the device S1 in the *Sensor Network1*. Upon receiving a CCN-lite data, the *Gateway* will scan its mapping table to find the short name. If there is a match the GW1 will extract the content from the CCN-lite data packet, generate a CCN data packet with the equivalent long name with the extracted content and forward it in the

Internet. When there is no match found in the mapping table the GW1 will discard the data packet.

Scenario 2: A user (U2) in the Internet is interested in receiving the temperature of the Computer Science building every 30mins. U2 will subscribe to the Content Descriptor (CD) */temperature/UNI/ComputerScience/Building1/*. Every 30mins, when BS1 has received the temperature of each room in the building, it will calculate the temperature of the building and publish a CCN-lite data with the name */temp/BS/Building1/Date/Time* (the parameter Date and Time should be replaced with the respective values). When the GW1 receives this packet it will check its mapping table and generate a CCN publication packet with the equivalent long name and the content and forward it to the subscribers in the Internet.

Scenario 3: The BS1 in the SN1 needs some content say the GPS location of the Mathematics building in the smart city. It generates a CCN-lite Interest with the ContentName */GPS/Location/UNI/Mathematics/Building2/* and forwards it to GW1. Please note that since this is an outbound traffic for content located outside the *Sensor Network* this should be indicated by setting any available bit in the Interest packet. For CCN-lite the EXCLUDE field can be used to indicate this. Upon receiving this Interest, the GW1 identifies it as an outbound traffic by inspecting the one bit field. The *Gateway* will only perform a protocol translation by generating a CCN Interest with the same name and forward it to the Internet. The content may be located in the Internet or also in another *Sensor Network*. The publisher either from the Internet or any other *Sensor Network* (with the help of its *Gateway* similar to scenario1) will reply with the content. Upon receiving a data packet the GW1 will again perform a protocol translation by extracting the content from the CCN data packet and generate the CCN-lite data packet with the content and forward it to BS1.

Scenario 4: In our scenario4 we consider the internal operation of *Sensor Network1*. The sensor devices are sensing temperature every 30mins. They are the publishers and will publish a packet every 30mins with the name */temp/id/room_no/Date/Time*, E.g. */temp/01/r1/03-11-16/12:30*. The BS1 has subscribed to the prefix */temp* and hence will receive the data sensed by all the sensors.

There are chances for the packets to get lost. If there is any packet loss, then the BS1 will generate a CCN-lite Interest with the specific name. E.g. If the packet from the S2 was lost then the BS1 will send a CCN-lite Interest with the name */temp/02/r2/03-11-16/12:30* at the next 30minute cycle when the S2 will wake up to sense the next temperature reading. For reliability reasons we require the sensors to store their latest three readings. When S2 receives an Interest, it will reply with the requested content. We suggest the device S2 to stay awake for at least 1RTT for the acknowledgement from BS1 to ensure that BS1 has received the data this time.

9 Discussions

9.1 Mobility

Mobility has become a norm in today's world and IoTs will be no exception to fulfill this need. A simple example is when a smart car in a smart city moves from building1 to building2. During this mobility, the smart car is detaching from the *Sensor Network* in building1 and is attaching to *Sensor Network* in building2.

In order to support mobility, the *Gateway* must handle the devices that move from one domain to another. The *Gateway* can either offer a Time To Live (TTL) during registration and/or offer a de-registration process. During de-registration the smart car sends an Interest with the de-registration request to *Gateway* and the *Gateway* responds with an acknowledgement. The car can choose to wait for the acknowledgement or not. When the car moves to another domain it again registers itself with the associated *Gateway* of the new domain. Even if the de-registration packet was lost, since the car has registered to anther *Gateway* the network will synchronize with the routing updates.

9.2 Security

9.2.1 Security in IoT

IEEE 802.15.4 provides the capability for some link-layer security. The authors in IETF standard [2] urge users to make use of it. A majority of the sensor devices in *Sensor Networks* are expected to operate within their networks. Acknowledging resource constraints in the IoT devices, we believe they should be equipped with the minimum level of security features necessary for their operation. The asymmetric key encryption is computationally complex for the *Sensor Networks* [18], so we suggest the devices in the *Sensor Networks* can use the features provided at the link layer for encryption and if additional security is desired then opt for symmetric key encryption.

Moreover, the *Sensor Networks* will benefit with the content based security provided by the ICN solutions. Interestingly, authors in [19] discuss Attribute Based Encryption (ABE) for ICN networks. However, the current ABE solution are heavy for the IoT networks. The Internet on the other hand can benefit greatly with ABE while the Gateway can assist in encryption and decryption of the content using light-weight security measures suitable for IoT networks.

9.2.2 Security in Internet

The devices in Internet are subject to more attacks compared to devices in the *Sensor Networks*. Moreover, the devices in Internet are relatively powerful compared to the devices in *Sensor Networks*. These devices are capable of handling complex computation and hence can opt for asymmetric key encryption. Although it is computational heavy, it is harder to decipher the content. Many ICN solutions provide security by securing the content unlike securing the communication link as in IP. E.g. NDN uses the digital signature of the publisher for authenticating all the content and also uses encryption for protecting private content.

10 Related Work

In this section we discuss some works focused on using ICN for IoT and broadly classify them into four categories: Architectural, Routing/Caching, Protocol and Security.

Architectural: Authors in [14] propose an initial high-level design for IoT using NDN architecture. They divide the NDN layer into two planes: Data plane and management & control plane. The data plane handles query/response while the control plane re-engineers the

current NDN routing plane. In [3], authors analyze the current TCP/IP solutions for supporting IoT. They argue that existing TCP/IP solutions are inefficient and propose that IoT can benefit by using the ICN. In [20], authors experiment with two ICN architectures MobilityFirst and NDN for IoT. They name it MF-IoT and NDN-IoT and compare their performance. Whereas in [21], authors propose to use ICN for IoT to realize service oriented communication. They use MobilityFirst as an example ICN architecture and modify it to support the service oriented communication in IoT.

Routing/Caching: Authors in [12] discuss the shortcomings of CCN protocol for IoT and propose a routing protocol with O(1) and almost no control traffic. They exploit the caching and data path in ICN to support the IoT requirements. They also show that CCN-lite uses 80% less memory compared to IP. Whereas in [13], authors study the benefit of caching with ICN for IoT in terms of energy consumption and bandwidth utilization in comparison with IP.

Protocol: Authors in [11] focus on a specific type of data retrieval pattern called Multi-Source Data retrieval. They say the current NDN architecture does not support this type of communication. In the proposed solution consumers use multi-source interest to retrieve data from multiple producers. They propose to delete the PIT entry based on parameter like Interest life time (TTL) instead of deleting when the data is received the first time. Whereas in [22], the authors study the potential for using the ICN based solutions for Wireless Sensor Area Networks (WSAN). They discuss about how ICN for WSAN's is different from ICN for Internet. They use flat names and continuous Interest to receive data sensed by multiple sensors as multiple sensors in WSAN sense the same data and respond to the Interest.

Securtiy: Authors in [23] propose a protocol for authenticating and authorizing new devices joining IoT mesh networks in ICN. They show 87% improvement in communication and 66% improvement in energy consumption compared ZigBee-IP solutions. While authors in [18] compare two approaches based on Asymmetric and Symmetric key encryptions for deploying new IoT devices in existing ICN deployments. They report that although the Asymmetric key based solutions incur lower traffic they impose higher demands on energy and time consumption.

11 Conclusion

We started with a discussion on IoT and their immanent explosive growth in near future. We discussed the shortcomings of current IoT designs and an introduction to ICN. We observed that ICN is more suitable for supporting IoT compared to the IP architecture. We also observed that IoT devices do not need the full NDN stack and can work with lighter versions like CCN-lite/NDN-lite. We discussed in detail the importance and requirements for incorporating *Sensor Networks* into the Internet, thus paving a way for them to join the IoT family. We analyzed various requirements for such an architecture to integrate the *Sensor Networks* and proposed *ISI* architecture with *Gateways*. We described in detail the responsibilities of such a *Gateway*. We further proposed a naming schema and communication protocol along with some possible mobility and security considerations for IoT networks. With the help of use cases we described the functionality of *ISI* architecture. As part of future work we intend to develop and demonstrate a working prototype of the proposed *ISI* architecture.

Acknowledgment

This work was supported by the joint EU H2020/NICT ICN2020 Project (Contract No. 723014. and NICT No. 184).

References

[1] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 78–82.

[2] G. Montenegro and N. Kushalnagar, "Transmission of ipv6 packets over ieee 802.15. 4 networks," RFC 4944, September 2007.

[3] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in iot networking via tcp/ip architecture," NDN Project, Tech. Rep. NDN-0038, Tech. Rep., 2016.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *CoNEXT*, 2009.

[5] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan, "COPSS: An Efficient Content Oriented Pub/Sub System," in *ANCS*, 2011.

[6] ——, "SAID: A Control Protocol for Scalable and Adaptive Information Dissemination in ICN," in *ICN*, 2016.

[7] S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "ORICE: An Architecture for Object Resolution Services in Information-Centric Environment," in *LANMAN*, 2015.

[8] "CCN-lite," http://www.ccn-lite.net/.

[9] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet*. John Wiley & Sons, 2011, vol. 43.

[10] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann, and K. K. Leung, "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 91–98, 2013.

[11] M. Amadeo, C. Campolo, and A. Molinaro, "Multi-source data retrieval in iot via named data networking," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 67–76.

[12] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the iot: experiments with ndn in the wild," *arXiv preprint arXiv:1406.6608*, 2014.

[13] J. Quevedo, D. Corujo, and R. Aguiar, "A case for icn usage in iot environments," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 2770–2775.

[14] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named data networking for iot: an architectural perspective," in *Networks and Communications (EuCNC), 2014 European Conference on*. IEEE, 2014, pp. 1–5.

[15] L. Zhang, D. Estrin, J. Burke, V. Jacobson, and J. Thornton, "Named Data Networking (NDN) Project," PARC, Tech. Report NDN-0001, 2010.

[16] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 3, pp. 2–13, 2012.

[17] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 269–284, 2016.

[18] M. Enguehard, R. Droms, and D. Rossi, "Poster: On the cost of secure association of information centric things," in *ICN*, 2016.

[19] A. M. Malik, J. Borgh, and B. Ohlman, "Attribute-based encryption on a resource constrained sensor in an information-centric network," in *ICN*, 2016.

[20] S. Li, Y. Zhang, D. Raychaudhuri, and R. Ravindran, "A comparative study of mobilityfirst and ndn based icn-iot architectures," in *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on*. IEEE, 2014, pp. 158–163.

[21] J. Chen, S. Li, H. Yu, Y. Zhang, D. Raychaudhuri, R. Ravindran, H. Gao, L. Dong, G. Wang, and H. Liu, "Exploit icn for realizing service-oriented communication in iot," *IEEE Communication Magazine (submitted for publication)*, 2016.

[22] N.-T. Dinh and Y. Kim, "Potential of information-centric wireless sensor and actor networking," in *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*. IEEE, 2013, pp. 163–168.

[23] A. Compagno, M. Conti, and R. Droms, "Onboardicng: a secure protocol for on-boarding iot devices in icn," in *ICN*, 2016.