Wireless Pers Commun manuscript No.
(will be inserted by the editor)

ICN as Network Infrastructure for Multi-Sensory Devices

Local Domain Service Discovery for ICN-based IoT environments

José Quevedo · Carlos Guimarães · Rui Ferreira · Daniel Corujo · Rui L. Aguiar

Abstract Information Centric Networking (ICN) is an emerging research topic aiming at shifting the Internet from its current host-centric paradigm towards an approach centred around content, which enables the direct retrieval of information objects in a secure, reliable, scalable, and efficient way. The exposure of ICN to scenarios other than static content distribution is a growing research topic, promising to extend the impact of ICN to a broader scale. In this context, particular attention has been given to the application of ICN in Internet of Things (IoT) environments. The current paper, by focusing on local domain IoT scenarios, such as multi-sensory Machine to Machine (M2M) environments, discusses the challenges that ICN, particularly Interest-based solutions, impose to service discovery. This work proposes a service discovery mechanism for such scenarios, relying on an alternative forwarding pipeline for supporting its core operations. The proposed mechanism is validated through a proof-of-concept prototype, developed on top of the Named Data Networking (NDN) ICN architecture, with results showcasing the benefits of our solution for discovering services within a collision domain.

Keywords Information Centric Networking · Internet of Things · Machine to Machine · Service Discovery

J. Quevedo (✉) · C. Guimarães · R. Ferreira · D. Corujo · R. L. Aguiar
Universidade de Aveiro, Instituto de Telecomunicações
Campus Universitário de Santiago, 3810-193, Aveiro, Portugal
E-mail: {quevedo;cguimaraes;rferreira;dcorujo}@av.it.pt, ruilaa@ua.pt

1 Introduction

The coupling of networking communication capabilities with devices of heterogeneous characteristics (e.g., sensors, actuators) has motivated different actors (ranging from academia, service providers, manufacturers, to operators) into the development of solutions that collectively are building the Internet of Things (IoT). The possibility to remotely make use of the sensing and actuating capabilities of such devices, turning them into communicating and processing platforms, has prompted the appearance of different "smart scenarios" [17]. The IoT, by providing a connection between the physical and digital worlds, has generated added value and set off a continuously growing number of connected devices (7.3 billion Machine-to-Machine (M2M) networked devices are expected by 2018 [11]). The associated information exchanges has raised connectivity concerns at different levels. This connectivity explosion has placed a new set of stringent requirements over the underlying networking fabric (e.g., scalability, energy efficiency, self-organization, semantic interoperability, privacy and security), thus highlighting the need for novel ideas and solutions, able to cope not only with these requirements, but also granting the capability to better face future challenges [28].

On a parallel development, over the last few years, we have been also witnessing the emergence of novel networking paradigms, such as Information Centric Networking (ICN) [2, 29]. ICN's networking operation is centred around content, moving away from the host-centric approach of the current Internet. This novel connection paradigm, unlike the original underlying architecture of the Internet, intrinsically supports advanced mechanisms, such as security, mobility support and efficient caching. As it might be expected, these ICN features, along with the possibility of expanding its range

of scenario applications still at the design stage [18], have encouraged the utilization of the ICN concepts for addressing IoT challenges [4, 26, 31]. Moreover, exposing ICN mechanisms to different scenarios not only contributes to its own development, but can actually provide new solutions for issues that challenge current Internet technologies.

Efficient device and service discovery has proved to be a complex and dynamic aspect of IoT scenarios [10]. In these scenarios, devices mainly rely on ad-hoc connections and protocols for communication; therefore a third-party infrastructure may not be always available to assume the discovery role. Different protocol stacks address this issue at different layers. For example, Bluetooth performs discovery at Layer 2 using broadcast messages, while IP-based protocols like Zeroconf [9] use multicast/broadcast for decentralised addressing and local domain discovery. In ICN networks, service discovery is a relatively new topic, and, to our knowledge, most of the prior work is focused on infrastructured networks, where a dedicated node executes the discovery functions, acting as a centralised server that aggregates discovery results. However, these approaches are not viable under ad-hoc protocols like WiFi-Direct or Bluetooth, or in mobility scenarios where nodes cannot take for granted the availability of a dedicated discovery broker.

Therefore, the aim of this paper is to contribute to the use of ICN protocols within local connectivity IoT scenarios (e.g., multi-sensory M2M environments) by extending existing ICN solutions with discovery capabilities. Concretely, we discuss how local area discovery can be designed for Interest-based ICN protocols (e.g., Named Data Networking (NDN) [30], Content Centric Networking (CCN) [16]) without depending on a dedicated infrastructure and propose a discovery mechanism that relies in Layer 2 broadcast protocols and supports both reactive and proactive operation modes. In doing so, Interest-based ICN nodes were provided with the capability to listen and to broadcast unsolicited ICN messages within the local network, by means of a novel alternative forwarding pipeline for local area communication. Finally, a proof-of-concept prototype of the solution was implemented on top of NDN and evaluated in two distinct deployments environments.

The remainder of this paper is organised as follows: Section 2 introduces the scenarios that motivate this work and defines our problem statement. Section 3 briefly introduces ICN concepts and provides an overview of previous works related to our proposal. Section 4 details the proposed solution, which is implemented as a proof-of-concept prototype, as described in section 5, and later assessed in section 6, where experimental results are presented. Finally, conclusions are drawn in section 7.

2 Scenario and Problem Statement

This section describes, by presenting a motivating scenario, the main problem we address on this work.

2.1 Motivating Scenario - Smart Farming

Alice's farm includes a large area of natural meadows where the cattle roam for optimum health and well-being. Each animal has been provided with an intelligent collar coupled with a set of sensors that report relevant information about the animal (e.g., identification, health indicators, location). Besides the daily data collection which takes place at the stalls, Bob, a regulatory veterinary, occasionally wanders around the meadows, in order to electronically check the health condition of the animals. In doing so, he first checks the information provided by their sensors, and in case of anomalies, treats them. On his way through the fields, and to ensure that he does not miss an animal, he uses an application to launch queries for sensors providing the desired information, and to listen for sensors announcing themselves. So, whenever Bob stays within the range of an animal, he is able to discover how to access the information provided by the sensors in its collar.

In this scenario, devices and services for the different animals are discovered using both reactive and proactive approaches. The former requires a request to be sent for discovering the services of each animal device, while the later allows services to proactively announce themselves.

2.2 Problem Statement

The advertisement and discovery of services can be used by clients to discover available service providers. Discovery protocols can be reactive (i.e., polling), proactive (i.e., spontaneous announcements) or hybrids (i.e., both reactive and proactive). In IoT deployments involving a large amount of data producers, supporting hybrid discovery is a critical aspect for the deployment to scale. However, while reactive protocols match the synchronous workflow seen in ICN, proactive protocols do not rely on polling for updates, instead information is usually broadcast to all nodes within the local network. Consequently, existing service discovery solutions for Interest-based ICNs are mainly limited to a reactive approach in which consumers interested in a particular

service have to ask for possible providers. Moreover, current local area discovery solutions are based in unicast communications between nodes and cannot leverage the broadcast nature of some media (e.g., WiFi). Our target is to address dynamic IoT scenarios, where the use of brokers may not be possible and a hybrid solution (both reactive and proactive) may be more suitable than a strictly reactive approach.

3 Background

This section, presents the main requirements for service discovery in IoT environments along with the fundamental aspects related to the ICN concept. Special consideration is given to the application of ICN in IoT environments, as well as existing approaches for service discovery on these Future Internet architectures.

3.1 Service discovery for IoT environments

As previously stated, IoT environments are characterized by a massive amount of heterogeneous nodes with disparate communication and computation resources targeted by different applications. Besides, typical IoT scenarios are highly dynamic, involving physical mobility, radio duty cycles, low power and lossy environments. As a result, the already mature discovery concepts from traditional networks are not easily applicable to the IoT and efficient service discovery in these environments remains a challenge. Ensuring the availability, scalability, interoperability for an efficient and effective service discovery requires high levels of automation (e.g., self-configuring, self-managing, self-optimizing).

Although centralized solutions ease the management of service registries, ensure their consistency and provide fast lookup mechanisms, relying in decentralized solutions and allowing the proactive advertisement of services are key elements for increasing the scalability of the solution in IoT environments. Additionally, in order to ensure interoperability among the massive amount of heterogeneous devices and applications, it is necessary to provide meaningful service descriptions (e.g, functionality, scope, behaviour, QoS) as well as flexible description matching algorithms (e.g., algorithms relaying on semantic similarity [19]). Ensuring the security and privacy of the pervasive and sensible information commonly exchange in IoT scenarios and applications (e.g., smart healthcare, logistics, transportation) are other major challenges associated to IoT discovery solutions. Finally, discovery systems should account for constant changes in the topology, keeping the information updated and ensuring load-balancing and fault tolerance.

A comprehensive survey on service discovery approaches is provided in [22], where authors define the prime criteria that need to be fulfilled for an autonomic service discovery. Analysed solutions are categorized according to: (i) its level of decentralization (i.e., centralized, distributed or decentralized), and (ii) its matchmaking reasoning level (i.e., syntactical, hybrid or semantic).

The different challenges we have showcased in this section has been the focus of recent research on discovery solutions for IoT environments. For example, in [10], authors propose a Service Discovery solution which relies on ZeroConf mechanisms and P2P technologies for integrating discovery mechanisms in both local and large scale. Authors in [15], use a fully distributed opportunistic approach in order to optimise the discovery of services offered by constrained nodes. Their solution leverages the broadcast nature of the wireless channel to optimise discovery tasks by transmitting messages using link-layer broadcasts to all neighbours which will cooperatively make the next decision. Finally, ZigBee's discovery protocol [27] defines both device and service discovery. Devices may be searched using either broadcast or unicast messages and ZigBee coordinators and routers respond with its own address as well as the ones from its child nodes. Service discovery requests are sent by means of a broadcast message, and any node providing the requested services response back, in an unicast manner, to the requesting node.

3.2 Interest-based ICNs

Despite existing ICN solutions share some core concepts of this novel paradigm (e.g., information oriented communication, content based security, in-network caching), different implementations follow different design choices (e.g., communication model, naming principles, routing and forwarding). In this work we focus on Interest-based ICN solutions.

Interest-based ICNs propose a communication model driven by the information consumers, and based on the exchange of request and response packets (i.e., Interest and Data packets, respectively). A name, contained in both types of packets, is used to identify the content being addressed. Figure 1 illustrates the communication process in these architectures, along with the main element composing an Interest-based ICN node. Consumers initiate the communication by issuing an Interest. Interests are forwarded, towards an entity holding the content, according to information stored in the Forwarding Information Base (FIB) and following

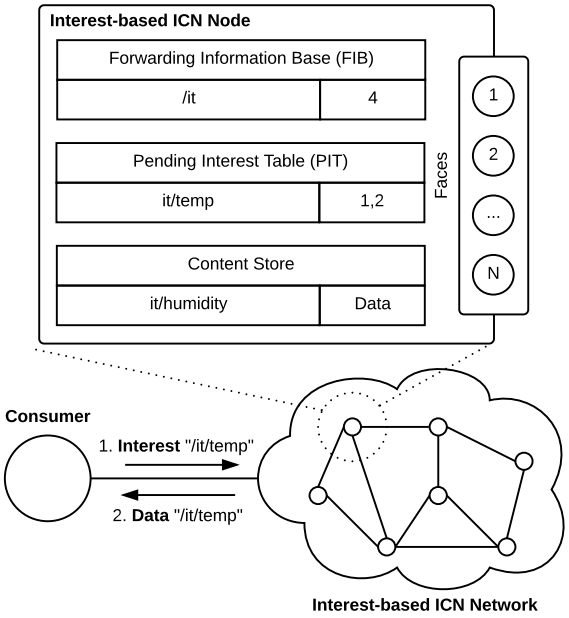

Fig. 1: Forwarding in Interest-based ICN networks

the configured forwarding strategy. Nodes maintain a Pending Interest Table (PIT) with information about outgoing forwarded requests (e.g., content name and incoming faces). Subsequent request for the same content are aggregated in the PIT (i.e., adding an incoming face to the PIT entry). Data is then routed back using the reverse request path based on the state information stored in the PIT. Upon the forwarding of a Data packet, the Interest is considered as satisfied and the corresponding PIT entry is removed (i.e. Data consumes Interest). Content objects can be cached in the Content Store (CS). Content objects are signed by the producers, ensuring both integrity and authenticity of the content.

3.3 ICN meets IoT

The alleged suitability of applying ICN concepts to IoT scenarios has triggered an increasing interest of the research community in exploring such an approach. Particularly, the Information-Centric Networking Research Group (ICNRG)[1] of the Internet Research Task Force (IRTF) has identified IoT as a baseline scenario where the use of ICN, as an underlying communication paradigm, could bring significant advantages compared to existing Internet protocols [18]. In enabling the deployment of current and envisioned IoT scenarios over ICN architectures, it is also expected to contribute to the development of ICN, thus opening a whole new set of scenarios which are not feasible under the current Internet architecture.

A detailed analysis identifying the main benefits, challenges and design choices for an efficient and scalable addressing of IoT scenarios from an ICN perspective is provided in [4, 26, 31]. Other works have addressed specific challenges of applying ICN concepts in IoT scenarios. For example, using long term Interests for enabling push-like communications [14]; lightweight alternatives to account for the memory and computational constraints of some IoT devices [24]; authenticated interest and encryption based access control for secure actuation [7] and sensing [8] in IoT-like environments; enabling data retrieval from multiple sources [3]; management aspects of IoT deployments over ICN [13]; impact of caching in energy and bandwidth efficiency [20]; information freshness [21]. Finally, authors in [6] perform an experimental analysis of the impact of ICN applied to IoT. Their work showcase the feasibility of using ICN in constrained devices and demonstrate that it can bring advantages over approaches based on 6LoWPAN/IPv6/RPL in terms of energy consumption, as well as RAM and ROM footprint.

3.4 Service Discovery in Interest-based ICN Architectures

CCNx[2] (version 1.0) specifications include a proposal of a Simple Service Discovery Protocol [25] based on the existence of a Service Discovery Broker responsible for managing the services within a Service Discovery Name Space. Services must be registered in the Service Discovery Broker and can be later discovered by clients. Replies to service discovery queries contain the names and additional metadata, for the services that have been admitted into the Service Discovery Name Space. In [19] a broker-based service discovery mechanism, based on NDN, is presented. The proposed mechanism leverages the use of a semantic matching mechanism for achieving a flexible discovery process. While these approaches are suitable for global IoT service discovery, dedicated entities are required to assume the discovery role and do not cover the infrastructure-less discovery, which could be largely benefit for local connectivity scenarios and better self-organization capabilities.

In [23], authors propose a CCNx prototype of an infrastructure-less service discovery mechanism. Their proposal included a Neighbour Discovery Protocol (NDP) and a Service Publish and Discovery Protocol

[1] https://irtf.org/icnrg

[2] www.ccnx.org

(SPDP). The NDP allows CCNx nodes to collect information about their locally reachable neighbour nodes, while the SPDP is responsible for receiving service registrations via API and for querying other SPDPs about available services. The querying process is based on a recursive hop-by-hop propagation of an Interest from one SPDP instance to another and also hop-by-hop aggregation of the response(s). While this approach shares our motivation, our proposal differs in that it enables proactive service advertisements.

4 Solution Overview

The current section details our proposal for a decentralized, hybrid service discovery mechanism for Interest-based ICN architectures.

4.1 Towards a proactive broker-less mechanism

From the previous section we learned that although there are some approaches that enable service discovery in Interest-based architectures, they lack some important features (e.g., proactive announcement of services and support for decentralized operation) relevant in targeting IoT/M2M scenarios, where protocol efficiency is a key aspect due to the common presence of resource constrained devices.

Supporting proactive and broker-less mechanisms for local service discovery in Interest-based ICN architectures requires two critical aspects to be addressed: (i) support for multiple source data retrieval, and (ii) support for a push-based communication model.

(i) Multiple source data retrieval:

As described in Section 3.2, Interest-based ICNs follow a pull-based communication model (i.e., every communication starts with an Interest which is consumed by a Data or otherwise expires). While Interest packets are routed, Data packets are forwarded based on PIT entries, which are deleted upon the reception of a corresponding Data packet. This communication model challenges the retrieval of pieces of information from multiple sources using a single Interest, which is the general case for a decentralized discovery procedure in a broadcast medium (i.e., a client wanting to discover available services sends a request and waits for the reception of multiple answers).

A possible solution is to handle this issue at the application layer by continuously reissue the same Interest but expressing in the *Exclude* field of the Interest the producers for which Data packets have been already received. The last Interest, after the content from all the producers has been already received, will timeout. However, this approach raises two main problems: (i) increased network overhead and delay, since for every Interest there is only one Data that reaches the client (ii) the overhead associated with Interests is continuously increased as more *exclude* related information is included.

A different approach, as proposed in [3], is to have longterm Interests, for which the corresponding PIT entries are not consumed by Data packets, but kept for the whole *Interest Lifetime*, thus maintaining the state information of a reverse path to be followed by multiple Data packets. The use of exclude filters is considered for the retrieval of lost Data packets, but this requires prior knowledge about the expected number of Data packets.

(ii) Push-based communication model:

In order to enable a proactive approach for service discovery (i.e., service providers periodically announce their services instead of just waiting for incoming queries), the support of a push-based communication model is required. However, as previously stated, Interest-based ICNs are designed to work only under a pull-based communication model.

A solution, used in [12], is for producers to send an Interest expressing their willingness to send Data. Consumers interested in the Data will then, in addition to the Data reply (potentially empty), send an Interest, allowing the producer to send the desired content.

In [14], authors explore the previously exposed idea of long term Interests, but now aiming to create a long lasting reverse path, allowing producers to push Data packets through that path toward interested consumers. After the expiration of the PIT entries, the consumers willing to keep this communication channel open can issue another long term Interest.

4.2 Key Concept: Elements for an alternative forwarding pipeline

Although some research works propose solutions for enabling these capabilities in Interest-based ICNs, they focus on forwarding issues rather than performance in constrained devices. In this context we propose an alternative forwarding pipeline for Interest-based ICNs. This new pipeline, which implementation details are given in section 5, will provide an additional forwarding path, based on rules other than PIT, FIB and CS matching, that leverages the broadcast nature of the media while satisfying the previously identified challenges by providing Interest-based ICN nodes with two additional capabilities:

1. Send Data messages into the local network (i.e., L2 collision domain) without having first received an Interest.
2. Receive Data messages without having to send an Interest message. Transitively, this capability also allows a node to receive multiple Data messages for a single Interest.

The scope of this work is intentionally limited to local networks, since that is a reasonable assumption in some IoT scenarios, and broadcast messages are not forwarded across the local collision domain. Notwithstanding, our proposal could be integrated with other solutions (e.g., [25], [19], [23]) for achieving a combined local/global solution as required for global scale IoT deployments. For example, such an integration could be achieved by allowing discovery brokers to discover the services available within its local domain scope (i.e., by leveraging our proposed mechanism) and to map them into globally addressable names (e.g., `/my-local-namespace/temp` maps into `/my-global-namespace/temp` and viceversa).

4.3 Local Area Service Discovery Mechanism

For clarity, in the following descriptions signalling workflows that take place inside the node are referred to as ***Internal***, while those that take place in the network between different nodes are tagged as ***External***.

We propose the design of a hybrid (both proactive and reactive) discovery mechanism that relies on the nodes' ability to receive/send unsolicited Data messages. The proposed discovery mechanism considers three types of applications: (i) Discovery Daemons, (ii) Services and (iii) Clients. Every node implementing this mechanism must have a Discovery Daemon running and may also contain one or more Service and/or Client applications. Additionally, it is assumed that the Discovery Daemon has been properly configured to leverage the novel alternative forwarding pipeline for listening/pushing unsolicited Data messages from/to the local network (e.g., during a bootstrap process triggered by the Discovery Daemon application). The reasoning behind the concept of a Discovery Daemon includes to avoid redundant queries by different applications, as well as to ensure a single node name.

Services are able to (un)register themselves in the local Discovery Daemon as shown in Figure 2. In doing so, Services send an Interest, which specifies the type of operation to perform (i.e., register or unregister), and also relevant information (e.g., name, inputs, outputs) about the provided service(s), properly encoded within

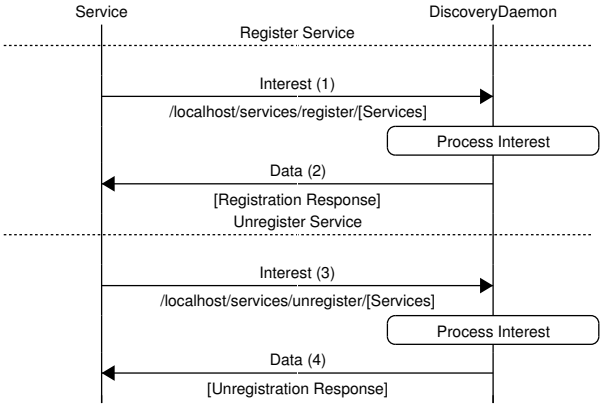

Fig. 2: Service – Discovery Daemon Communication ***(Internal)***

the name. The Discovery Daemon responds with a Data containing the result of the operation.

As shown in Figure 3, Discovery Daemons are responsible for the service discovery and announcement processes by exchanging information, within the local network, on behalf of client/services applications running in the node. During its bootstrap, a Discovery Daemon sends an initial query, *Interest (1)*, with the objective of checking the availability of a name to be associated to the node. If the Interest times out, it means that the name is available. On the other hand, the reception of a *Data (2)* packet means that the name is already in use by another Daemon and a different one must be chosen (this mimics the name collision detection mechanism seen in Zeroconf [9]). Additionally, a Discovery Daemon can query for services, *Interest (3)*, and consequently receive *Data (4)* packets from other Discovery Daemons containing the specifications of the services they provide and that satisfy the query. The Discovery Daemon can also operate proactively, by sending periodical announces, *Data (5)*, containing the information regarding the services being provided at the node. The announcement interval can be configured according to different policies (e.g., reducing network and energy overhead). This packet is received by the Discovery Daemons running in neighbouring nodes which, in turn, update their local information about remote services. This information is associated to an expiration time, which will be renewed through new incoming announcements or will otherwise expire. The Discovery Daemon may also use a "Bye" Data packet (*Data (6)*) for removing its services from the other Discovery Daemons before their expiration time.

Similarly, as depicted in Figure 4, Client applications query the local Discovery Daemon, *Interest (1)*, to find out the services offered within the local network,

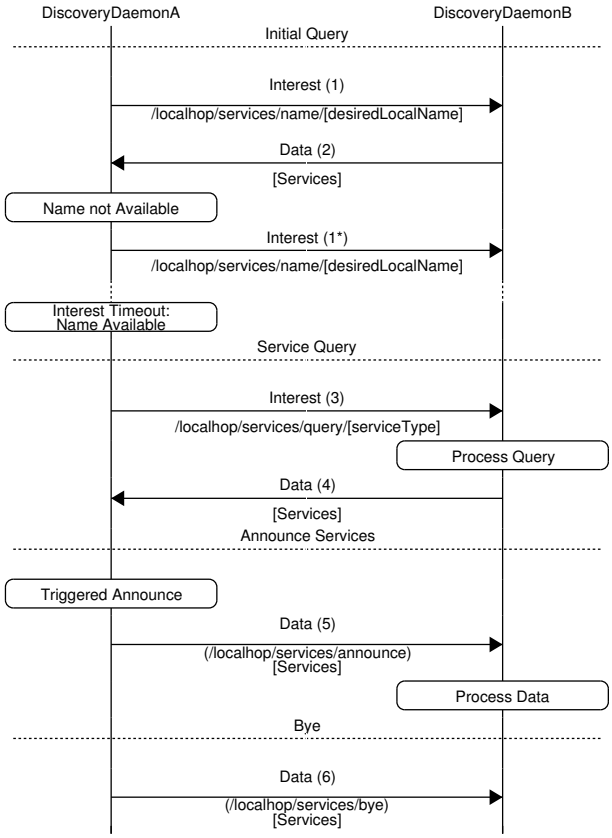

Fig. 3: Discovery Daemon – Discovery Daemon Communication ***(External)***

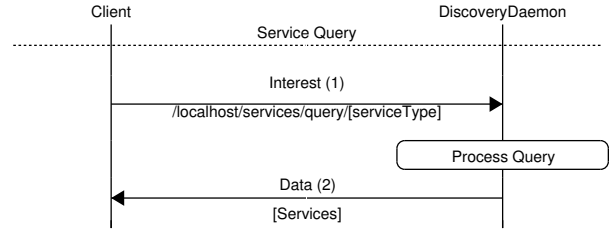

Fig. 4: Client – Discovery Daemon Communication ***(Internal)***

including the node itself. The Discovery Daemon replies with a *Data (2)* containing the information related to the relevant services. If the Discovery Daemon does not hold a valid answer for the query, it will perform a remote request for it, as previously described (Figure 3).

4.3.1 Security Considerations

ICN intrinsically supports content-based security on which each Data packet is secured, enabling data replication and preserving its security properties for the packet's lifetime. As such, we rely on this ICN mechanism for securing the proposed discovery protocol. ICN solutions generally consider different types of signature algorithm targeting application scenarios with different requirements in terms of tradeoff between security level and resource consumption. In section 6.2 we present an evaluation of the impact of these algorithms on top of constrained devices.

5 Proof-of-concept Prototype

A proof-of-concept prototype was implemented following the NDN architecture and basing its development on the NDN Platform[3] (version 0.3.2). We draft our solution on top of the NDN Platform as it is a reference open source platform for Interest-based ICNs, which is up to date with the latest developments of the NDN architecture and allows us to evaluate the impact of a full NDN stack on top of both constrained and unconstrained devices.

The reference NDN implementation considers a node to be composed by different *Faces* (i.e., a generalization of interface that may represent either a physical interface, an overlay tunnel or a UNIX-domain socket to a local application). The different Faces communicate with each other through the NDN Forwarding Daemon (NFD) [1], which maintains internal data structures such as CS, PIT, and FIB, and implements the packet processing logic.

As part of the prototype development process, besides the implementation of the Discovery Daemon, Service and Client applications, both the ndn-cxx and the NFD implementations were extended to support the novel Alternative Forwarding Pipeline (AFP). This pipeline provides an additional forwarding path for the intra-node face communication through the NFD, based on rules other than PIT, FIB and CS matching, (e.g., packets with prefix /localhop/services incoming from face A will always be forwarded to face B).

Figure 5 illustrates the forwarding process of a NDN node and highlights with dashed lines the extensions introduced to support the new pipeline. Any packet arriving to a Face, besides following its normal path, will also be checked against a new Alternative Forwarding Table (AFT). An AFT entry is composed by a name filter, a packet type, a list of incoming faces and a list of outgoing faces. Any packet matching an AFT entry will be forwarded to the outgoing face(s) therein specified. Therefore, enabling alternative forwarding for a given Face (i.e., include the Face in the list of outgoing faces of an AFT entry), will enable the reception of unsolicited packets on that Face. Consequently, enabling alternative forwarding for a Face associated to a

[3] `http://named-data.net`

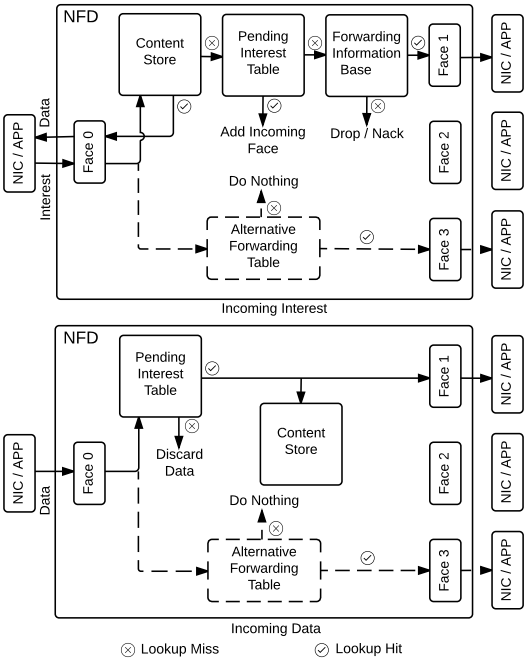

Fig. 5: Extended Forwarding Process of a NDN Node

Network Interface Controller (NIC) will push any NDN packet matching the AFT entry into the L2 collision domain.

The proposed pipeline is managed through the API provided by the NFD (i.e., an exchange of Interest/Data control commands), which was also extended with a new module for that purpose. Following the NFD specifications, the */localhost* and */localhop* namespaces are enforced to ensure that packets being exchanged cannot leave the node and the local network, respectively.

By defining an AFT rule involving input and output faces corresponding to different NICs, the scope of the alternative pipeline concept could be extended beyond the local domain. However, on the current paper, as far as it is sufficient for enabling local area service discovery, we limit the impact of this mechanism to the local domain. Also, the proposed modifications are intended only to the nodes themselves and not to the NDN routers. Therefore, in the current proposal, we only consider AFT entries with at most one NIC face. Forwarding beyond the local scope should be handled at the application layer or otherwise through the traditional NDN forwarding pipeline.

Although we draft our solution for local domain service discovery on top of NDN, the main concepts are equally applicable to similar ICN architectures.

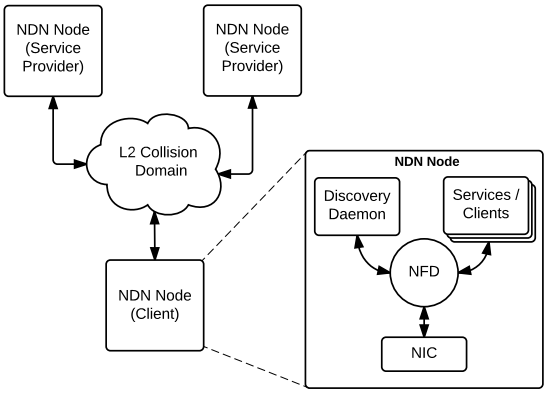

Fig. 6: Evaluation Scenario

6 Evaluation

The current section presents an evaluation of our proposal conducted by deploying the proof-of-concept prototype in two different experimental environments: (i) Constrained and (ii) Unconstrained.

To validate our proposal, we focused on two parameters: (i) the service time (i.e., the amount of time elapsed from the moment when the request is sent, up to the reception of the desired response) and (ii) the overhead introduced in the network by the discovery protocol.

The evaluation scenario, as shown in Figure 6, is composed by three nodes, two service providers and one client, all of them connected to the same L2 collision domain. This setup allowed us to verify the correct operation of the prototype in multi-source environments. The two environments differ in the hardware/technologies used for instantiating the scenario. The first involved the use of virtual machines (single core 3.33GHz virtualised CPU with 2GB of RAM) hosted in an OpenStack Platform and connected through Gigabit Ethernet. The second is based on Raspberry Pi Model B devices connected via IEEE 802.11g interfaces. These two scenarios will allow us to evaluate the behaviour of our prototype and the NDN stack in general, considering different device capabilities.

6.1 Service time analysis

We evaluated the service time for the three main operations of our solution: register service, unregister service and service query (Figures 2, 3 and 4). The number of services being processed in each evaluation ranged from 1 to 10 (with a resolution of 1 service) to analyse its impact on the service time. Two different approaches to request the (un)registration of services were stud-

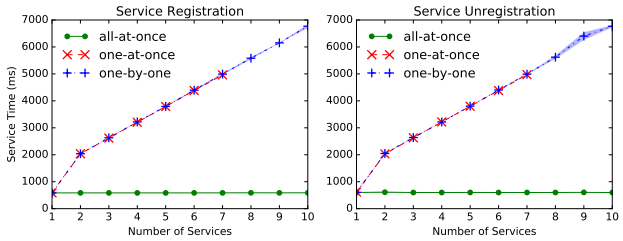

Fig. 7: Constrained environment evaluation

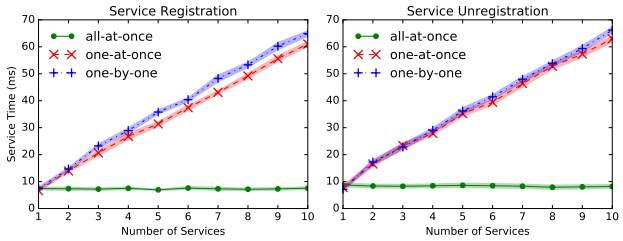

Fig. 8: Unconstrained environment evaluation

ied: 1) all services in a single request (*all-at-once*), and 2) one service per request (*one-per-request*). This last approach was also divided into two different strategies depending on whether the requester waits (*one-by-one*) or not (*one-at-once*) for an answer before sending the next request. In all cases, the amount of time considered is the total time elapsed from the moment when the first request is sent, until the reception of the last response. All evaluations were run 50 times and a 95% confidence interval was calculated.

Results (Figures 7 and 8) for the service registration and unregistration events in both evaluated environments were, as expected, quite similar. The results when considering the different *one-per-request* strategies were also quite similar. However, in the *one-at-once* strategy, the Constrained environment was unable to handle more than seven concurrent (un)registration requests, beyond this number some requests remained unanswered. The reason behind it is that the time that would be required for responding to all the requests exceeds the Interest Lifetime (5000ms for the Constrained environment) and consequently some Interests expire before they can be answered. Using the *all-at-once* approach showed no considerable increase on the service time as the number of services is increased. On the other hand, increasing the number of services in the *one-per-request* approaches resulted in a linear increase of the service time.

A comparison among the results from the two different environments shows that the service time on the Constrained environment is more than 100 times higher than its equivalent in the Unconstrained environment.

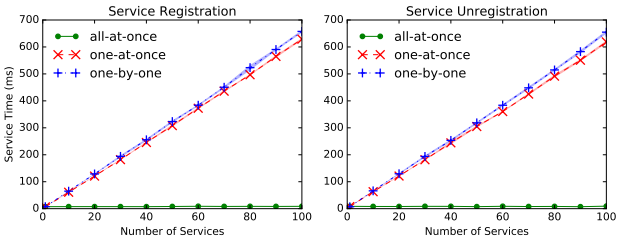

Fig. 9: Unconstrained environment extended evaluation

For simplicity, the service query response time was studied in the Client-DiscoveryDaemon interface. For both studied environments the service time showed almost no variation with respect to the amount of services being processed (approximately 3 *ms* and 565 *ms* for the Unconstrained and Constrained environments respectively).

The previous analysis was extended from the previous maximum of 10 services to a maximum of 100 services but limited to the Unconstrained environment. Results are shown in Figure 9, and demonstrate that our solution, when considering the *all-at-once* approach scales in terms of number of services. The *one-per-request* approaches, as expected, keep the growing tendency and are therefore not recommendable for a high number of services.

6.2 Impact of ICN security mechanisms

The key point for variable security levels in ICN is the level of encyption that is applied to the ICN packets signing operations. A detailed analysis on the origin of the high values of service time for the Constrained environment, as compared with those for the Unconstrained one, revealed that most of the time was associated to Data packets signing operations. Consequently, in addition to the already studied RSA-2048 signature (default algorithm of the NDN implementation), we extended our previous evaluation for Constrained environments to study alternative signature types considered by NDN, namely ECDSA-256 and Digest Only (SHA-256). Results are shown in Figure 10, evidencing the impact that the use of each signature type has on the service time, with associated savings ranging from 69% to 83% for ECDSA-256 and from 96% to 98% for SHA-256.

Additionally, an assessment of the processing time for the signature and verification processes in a Raspberry Pi Model B was conducted by using the benchmark tools of two different cryptography libraries:

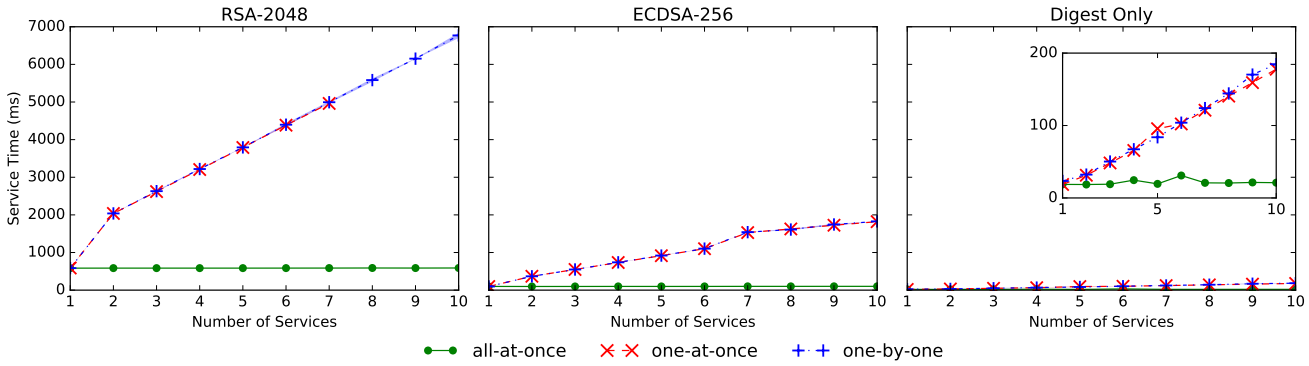

Fig. 10: Impact of different signature algorithms in Service Registration time

Table 1: Cryptography Libraries Time Benchmark

Algorithm	Library	Signature [*ms*]	Verification [*ms*]
RSA-2048	Crypto++	237.68	2.58
	OpenSSL	75.5648	2.352
ECDSA-256	Crypto++	27.64	62.48
	OpenSSL	3.4	15.2

Crypto++ v5.6.3[4] (used by ndn-cxx) and OpenSSL v1.0.1e[5]. The assessment considered the signature algorithms used by NDN, with results regarding times for signature and verification processes being shown in Table 1. Results show that the OpenSSL library outperforms the Crypto++ library (likely due to platform specific optimizations), notwithstanding it remains a time consuming process requiring further attention.

6.3 Network overhead analysis

The overhead analysis was limited to the packets exchanged over the network (i.e., skipping packets exchanged over internal UNIX-domain faces). The initial query Interest[6] for our implementation was determined to be 79 bytes, while the discovery Interest[7] was 92 bytes. As in the case of the Data packet containing the services, Figure 11 shows the size of the Data Packets as a function of the amount of services it contains. In this figure, the hypothetical curve where each service announcement is sent in an individual packet, as well as the saves associated with the aggregation of services into a single Data packet (i.e., the percentage of the total size of sending single service announcements,

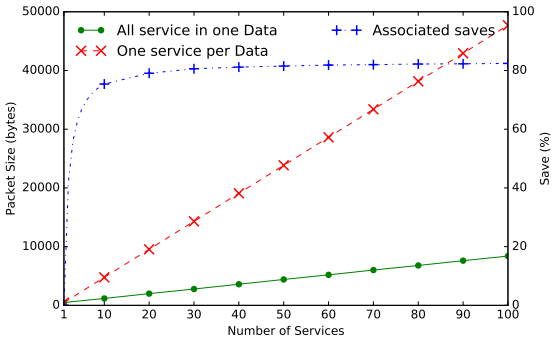

Fig. 11: Overhead

that can be saved by performing aggregation), have also been represented.

7 Conclusions and discussion

With the increased interest for ICN in IoT/M2M environments, the ICN protocol stacks are revisiting the challenges of efficient service discovery. The natural pull-based model seen in ICN architectures is meant for routing optimisations based on content addressing. However, in IoT scenarios, nodes may only require local connectivity through service discovery, specially in decentralised or mobile scenarios where no infrastructure can be assumed.

This is a specific problem for Interest-based ICN architectures because they generally assume a data flow in which Interests packets are consumed by Data packets, thus neglecting multi-source content retrieval and push-based communication scenarios. Our paper addresses these issues, in the context of local network communication, for achieving efficient IoT local service discovery in these architectures. We proposed a discovery protocol that can be both reactive and proactive and that better leverages the broadcast nature of wireless media,

[4] `www.cryptopp.com`
[5] `www.openssl.org`
[6] Interest name: `/localhop/services/name/nodeX`
[7] Interest name: `/localhop/services/query/DummyDataProvider`

extending the NDN reference implementation with the notion of an alternative forwarding pipeline to support such capabilities. As a result, NDN nodes become able to send and receive Data messages without a matching Interest.

As a proof of concept of our solution, a prototype of the discovery protocol was developed and tested experimentally in two distinct deployments environments: one using Raspberry Pi nodes communicating over WiFi, and another using regular virtualised nodes connected using Ethernet. Our implementation and choice of discovery strategies displayed enough flexibility for different applications or services to parametrise configuration based on their needs (e.g., some applications might need reduced service times while others prefer reduced payload lengths).

Experimental results support the viability of using our service discovery solution. However, the experimentation of the full NDN stack on top of a fully functional operating system running on constrained devices, such as the Raspberry Pi Model B, showed some limitations. Based on this fact, further work on this direction should consider the use of other OS, such as RiotOS[8] [5] and different named data networking approaches, such as CCNLite[9].

Additionally, there is room for improvements regarding cryptographic mechanisms, namely the need for proper choices of algorithms and platform specific optimizations. The choice of a proper cryptographic algorithm could depend on the target scenarios, and the roles played by the IoT devices. For example, constrained nodes assuming a consumer role could benefit from the use of RSA signing (fast to verify), while ECDSA schemes could be more effective in a producer role (fast signing). Moreover, using just hashing may be a valid approach in discovery scenarios where provenance protection is not a requirement.

Although our approach for service discovery targeted local domain scenarios, it could be applicable beyond the local scope by integrating it with other solutions. Moreover, the applicability of the proposed alternative forwarding pipeline implementation is not exhausted to service discovery scenarios, but may be considered as a general purpose tool for other applications (e.g., packet sniffing application).

Acknowledgements This work was supported within the scope of R&D Unit 50008, this work was also financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER - PT2020 partnership agreement) with ref. no. UID/EEA/50008/2013. FCT Grant SFRH/BD/96553/2013

[8] `www.riot-os.org`

[9] `www.ccn-lite.net`

References

1. Afanasyev, A., Shi, J., Zhang, B., Zhang, L., Moiseenko, I., Yu, Y., Shang, W., Huang, Y., Abraham, J.P., DiBenedetto, S., et al.: NFD developers guide. Tech. rep., Technical Report NDN-0021, NDN Project (2014)
2. Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B.: A survey of information-centric networking. IEEE Communications Magazine **50**(7), 26–36 (2012). DOI 10.1109/MCOM.2012.6231276
3. Amadeo, M., Campolo, C., Molinaro, A.: Multi-source Data Retrieval in IoT via Named Data Networking. In: Proceedings of the 1st International Conference on Information-centric Networking, INC '14, pp. 67–76. ACM, New York, NY, USA (2014)
4. Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera, A., Aguiar, R.L., Vasilakos, A.V.: Information-Centric Networking for the Internet of Things: Challenges and Opportunities. IEEE Network Magazine (2015)
5. Baccelli, E., Hahm, O., Günes, M., Wählisch, M., Schmidt, T.: RIOT OS: Towards an OS for the Internet of Things. In: Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, pp. 79–80 (2013). DOI 10.1109/INFCOMW.2013.6970748
6. Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T.C., Wählisch, M.: Information Centric Networking in the IoT: Experiments with NDN in the Wild. In: 1st ACM Conference on Information-Centric Networking (ICN-2014), pp. 77–86 (2014). DOI 10.1145/2660129.2660144
7. Burke, J., Gasti, P., Nathan, N., Tsudik, G.: Securing Instrumented Environments over Content-Centric Networking: the Case of Lighting Control and NDN. In: IEEE NOMEN Workshop (2013)
8. Burke, J., et al.: Secure Sensing over Named Data Networking. In: IEEE Network Computing and Applications (NCA), pp. 175–180 (2014)
9. Cheshire, S., Krochmal, M.: Multicast DNS. RFC 6762 (Proposed Standard) (2013). URL `http://www.ietf.org/rfc/rfc6762.txt`
10. Cirani, S., Davoli, L., Ferrari, G., Leone, R., Medagliani, P., Picone, M., Veltri, L.: A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things. IEEE Internet of Things Journal **1**(5), 508–521 (2014). DOI 10.1109/JIOT.2014.2358296
11. CISCO: Cisco Visual Networking Index: Global IP Traffic Forecast, 2014 - 2019. Tech. rep., CISCO (2015)
12. Corujo, D., Aguiar, R.L., Vidal, I., Garcia-Reinoso, J.: A named data networking flexible framework for management communications. IEEE Communications Magazine **50**(12), 36–43 (2012). DOI 10.1109/MCOM.2012.6384449
13. Corujo, D., Aguiar, R.L., Vidal, I., García-Reinoso, J., Pentikousis, K.: Research challenges towards a managed information-centric network of things. In: European Conference on Networks and Communications, EuCNC 2014, Bologna, Italy, June 23-26, 2014, pp. 1–5 (2014)
14. Dinh, N.T., Kim, Y.: Potential of information-centric wireless sensor and actor networking. In: Computing, Management and Telecommunications (ComManTel), 2013 International Conference on, pp. 163–168. IEEE (2013)

15. Djamaa, B., Richardson, M., Barker, P., Owens, I.: Discovery of Things: A Fully-Distributed Opportunistic Approach. 2015 IEEE 81st Vehicular Technology Conference (VTC Spring) pp. 1–5 (2015). DOI 10.1109/VTCSpring.2015.7145778
16. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT '09, pp. 1–12. ACM (2009). DOI 10.1145/1658939.1658941
17. Miorandi, D., Sicari, S., Pellegrini, F.D., Chlamtac, I.: Internet of things: Vision, applications and research challenges . Ad Hoc Networks **10**(7), 1497 – 1516 (2012)
18. Pentikousis, K., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., Eum, S.: Information-Centric Networking: Baseline Scenarios. RFC 7476 (Informational) (2015). URL `http://www.ietf.org/rfc/rfc7476.txt`
19. Quevedo, J., Antunes, M., Corujo, D., Gomes, D., Aguiar, R.L.: "On the application of Contextual IoT Service Discovery in Information Centric Networks ". Computer Communications pp. – (2016). DOI 10.1016/j.comcom.2016.03.011
20. Quevedo, J., Corujo, D., Aguiar, R.: A case for ICN usage in IoT environments. In: Global Communications Conference (GLOBECOM), 2014 IEEE, pp. 2770–2775 (2014)
21. Quevedo, J., Corujo, D., Aguiar, R.: Consumer driven information freshness approach for content centric networking. In: Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, pp. 482–487 (2014). DOI 10.1109/INFCOMW.2014.6849279
22. Rambold, M., Kasinger, H., Lautenbacher, F., Bauer, B.: Towards Autonomic Service Discovery A Survey and Comparison. In: 2009 IEEE International Conference on Services Computing, Section II, pp. 192–201. IEEE (2009). DOI 10.1109/SCC.2009.59
23. Ravindran, R., Biswas, T., Zhang, X., Chakraborti, A., Wang, G.: Information-centric networking based homenet. In: Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, pp. 1102–1108 (2013)
24. Ren, Z., Hail, M., Hellbruck, H.: CCN-WSN - A lightweight, flexible Content-Centric Networking protocol for wireless sensor networks. In: Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on, pp. 123–128 (2013). DOI 10.1109/ISSNIP.2013.6529776
25. Scott, G.: CCNx 1.0 Simple Service Discovery. Tech. rep., Computing Science Laboratory, Palo Alto Research Center (2014)
26. Shang, W., Bannis, A., Liang, T., Wang, Z., Yu, Y., Afanasyev, A., Thompson, J., Burke, J., Zhang, B., Zhang, L.: Named data networking of things (invited paper). In: 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), pp. 117–128 (2016). DOI 10.1109/IoTDI.2015.44
27. Specification, Z.: Zigbee standards organization. Document 053474r17, Jan **17**, 26 (2008)
28. Stankovic, J.: Research Directions for the Internet of Things. IEEE Internet of Things Journal **1**(1), 3–9 (2014). DOI 10.1109/JIOT.2014.2312291
29. Xylomenos, G., Ververidis, C., Siris, V., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K., Polyzos, G.: A Survey of Information-Centric Networking Research. IEEE Communications Surveys Tutorials **16**(2), 1024–1049 (2014). DOI 10.1109/SURV.2013.070813.00063
30. Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Crowley, P., Papadopoulos, C., Wang, L., Zhang, B., et al.: Named Data Networking. ACM SIGCOMM Computer Communication Review **44**(3), 66–73 (2014)
31. Zhang, Y., Raychadhuri, D., Grieco, L.A., Baccelli, E., Burke, J., Ravindran, R., Wang, G., Lindgren, A., Ahlgren, B., Schelen, O.: Requirements and Challenges for IoT over ICN. Internet-Draft draft-zhang-icnrg-icniot-requirements-00, Internet Engineering Task Force (2015). URL `https://tools.ietf.org/html/draft-zhang-icnrg-icniot-requirements-00`. Work in Progress