On the application of Contextual IoT Service Discovery in Information Centric Networks

José Quevedo*, Mário Antunes, Daniel Corujo, Diogo Gomes, Rui L. Aguiar

Universidade de Aveiro, Instituto de Telecomunicações, Campus Universitário de Santiago, 3810-193, Aveiro (Portugal)

Abstract

The continuous flow of technological developments in communications and electronic industries has led to the growing expansion of the Internet of Things (IoT). By leveraging the capabilities of smart networked devices and integrating them into existing industrial, leisure and communication applications, the IoT is expected to positively impact both economy and society, reducing the gap between the physical and digital worlds. Therefore, several efforts have been dedicated to the development of networking solutions addressing the diversity of challenges associated with such a vision. In this context, the integration of Information Centric Networking (ICN) concepts into the core of IoT is a research area gaining momentum and involving both research and industry actors. The massive amount of heterogeneous devices, as well as the data they produce, is a significant challenge for a wide-scale adoption of the IoT. In this paper we propose a service discovery mechanism, based on Named Data Networking (NDN), that leverages the use of a semantic matching mechanism for achieving a flexible discovery process. The development of appropriate service discovery mechanisms enriched with semantic capabilities for understanding and processing context information is a key feature for turning raw data into useful knowledge and ensuring the interoperability among different devices and applications. We assessed the performance of our solution through the implementation and deployment of a proof-of-concept prototype. Obtained results illustrate the potential of integrating semantic and ICN mechanisms to enable a flexible service discovery in IoT scenarios.

Keywords: Internet of Things, Information-Centric Networking, Context Information, Service Discovery, Semantic Similarity

*Corresponding author

Email addresses: quevedo@av.it.pt (José Quevedo), mario.antunes@av.it.pt (Mário Antunes), dcorujo@av.it.pt (Daniel Corujo), dgomes@av.it.pt (Diogo Gomes), ruilaa@ua.pt (Rui L. Aguiar)

1. Introduction

In the last few years, the coupling of networking communication capabilities and devices with disparate characteristics and capabilities (e.g., sensors, actuators) has prompted different actors (ranging from academia, to service providers, manufacturers and operators) into the development of solutions towards an Internet of Things (IoT). These solutions are able to remotely exploit the sensing and actuating capabilities of such devices and convey them into communicating and processing platforms, empowering different kinds of “smart” scenarios [1, 2]. The added value generated by bridging the physical and digital worlds has contributed to a continuously increasing massification of connected devices and generated information exchanges ([3] indicates 7.3 billion Machine-to-Machine (M2M) networked devices by 2018, globally), raising connectivity provisioning and operation concerns at all levels. The stringent new requirements placed over the underlying networking fabric to support this connectivity explosion have prompted the need for ground-breaking ideas and solutions, able not only to support these challenges, but also to confer the capability and flexibility to better face future challenges and requirements.

Information Centric Networking (ICN) [4, 5] is an emerging networking paradigm that has content at the centre of the networking functions, shifting from the current host-centric approach of the Internet. Moreover, unlike the current underlying architecture of the Internet, this new approach intrinsically couples its networking procedures with important supportive mechanisms, such as security, mobility support and efficient caching. These capabilities, along with the possibility of expanding its range of scenario applications at the design stage [6], have naturally brought the ICN and IoT concepts closer [7, 8], allowing the pursuit of ICN as an IoT-capable platform, while exposing it to new scenarios and contributing to its own development. Moreover, this approach can actually provide new solutions for open issues that plague current Internet mechanisms.

In the IoT, different devices/manufacturers specify their own structure for sharing information leading to information silos [9]. This has hindered the interoperability between different applications and the realization of more complex IoT scenarios. Moreover, efficient device and service discovery has proven to be a complex and dynamic aspect of IoT scenarios [10]. Therefore, in order to make information useful and to ensure interoperability among different applications, it is necessary to provide data with adequate and standardized formats, models and semantic description of their content (metadata), using well-defined languages and formats [1]. However, the lack of standards and the heterogeneity of formats for describing IoT content has triggered research on techniques to deal with unstructured information, where particular emphasis has been given to semantic similarity. The goal behinds its application is to enable the adoption of the IoT on a wide scale by allowing the proper identification of information with similar context, regardless of the vocabulary used therein [11].

The aim of this paper is thus to contribute to the deployment and usability of ICN protocols by extending existing solutions with semantic discovery capabilities. Consequently, we integrate and evaluate the unsupervised semantic similarity solution proposed in [12] with an ICN-based discovery mechanism developed on top of the Named Data Networking (NDN) architecture [13]. In doing so, some of the core concepts of [12] had to be further evolved and a novel service-query matchmaking interface was developed.

The remainder of this paper is organised as follows: Section 2 briefly introduces ICN concepts, contextualize its usage in IoT environments and provides an overview of previous work on service discovery and semantic matching techniques. Section 3 defines the problem statement. Section 4 details the proposed solution and section 5 discusses experimental results. Finally conclusions are provided in section 6.

2. Background and related work

In this section, we present the fundamental aspects related to the ICN concepts, with emphasis on Interest-based ICNs, along with the application of those concepts for service discovery and in IoT environments. Additionally the section presents some background on the main methods used for evaluating the semantic distance between two words, and concludes with some remarks regarding recent efforts to support Service Discovery in IoT environments.

2.1. Information-Centric Networking

Although existing ICN solutions share the 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 will focus on Interest-based ICN solutions. Interest-based ICNs (e.g., Named Data Networking (NDN) [13], Content Centric Networking (CCN) [14]) propose a communication model driven by the information consumers and based on the exchange of two packet types, i.e., Interest and Data. A name, contained in both types of packets, is used to identify the content being addressed. Requests (Interests) for a given piece of information are forwarded towards the producer(s) of the content according to the information stored in the Forwarding Information Base (FIB) and following a configured Forwarding Strategy. Nodes maintain a Pending Interest Table (PIT) for outgoing forwarded requests and map them to the network interface from where the corresponding requests have been received. 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). The nodes involved in the communication can cache both requests (through aggregation in the PIT) and content objects (in the Content Store (CS)). Content objects are signed by the producers, ensuring both integrity and authenticity of the content.

2.1.1. Information-Centric Networking for the Internet of Things

In the recent years, the research community has been witnessing an increasing interest on the application of the ICN concepts in addressing IoT scenarios. 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 underlying communication paradigm, could bring significant advantages compared to existing Internet protocols [6]. Some relevant works have provided a detailed analysis on addressing IoT scenarios from an ICN perspective, identifying the main benefits and challenges, along with some design choices aiming at an efficient and scalable realization of such technology integration [7, 8, 15].

Different research works have tackled particular challenges of enabling an ICN-based IoT. For example, enabling push-like communications through long lasting Interests [16]; lightweight alternatives to meet the memory and computational constraints of some IoT devices [17]; authenticated interest and encryption based access control for secure actuation [18] and sensing [19] in IoT-like environments; enabling data retrieval from multiple sources [20]; management aspects of IoT deployments over ICN [21], impact of caching in energy and bandwidth efficiency [22], information freshness [23].

Authors in [24], go one step further and provide an experimental analysis of the shortcomings of ICN applied to IoT. Their work showcase the feasibility of using ICN in constrained devices and show that it can bring advantages over approaches based on 6LoWPAN/IPv6/RPL in terms of energy consumption, as well as in terms of RAM and ROM footprint.

2.1.2. Service Discovery in ICN

PARC[2] included a description of a Simple Service Discovery Protocol [25] within the specifications of their latest release of CCNx[3] (version 1.0). The proposed scheme is 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 to the Service Discovery Name Space.

In [26], authors propose a CCNx prototype of an infrastructure-less service discovery mechanism. The proposal included two different protocols, a Neighbour Discovery Protocol (NDP) and a Service Publish and Discovery Protocol (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 an API and for querying other SPDPs about available services. The querying process is based on a recursive hop-by-hop propagation of an In-

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

[2] `www.parc.com`

[3] `www.ccnx.org`

terest from one SPDP instance to another and also hop-by-hop aggregation of the response(s).

2.2. Semantic Distance Estimation

Semantic distance is a measure of proximity between two units of language, in terms of their meaning. For example, the nouns *"temperature"* and *"heat"* are closer in meaning than the nouns *"temperature"* and *"acceleration"*. In this context, semantic distance estimation methods can be divided in two classes: *(i)* Lexical-resource-based measures of concept-distance, and *(ii)* Distributional measures of word-distance.

Lexical-resource-based measures of concept-distance rely on the structure of a knowledge source, such as WordNet [27], to determine the distance between two concepts. In the WordNet database, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets). Synsets express different concepts and are interlinked by means of conceptual-semantic and lexical relations. Although WordNet resembles a thesaurus, as it groups words together based on their meanings, there are some important differences. First, WordNet not only interlinks word forms (strings of letters), but also specific senses of words. As a result, words that are found to be on the proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity. Several authors have proposed semantic measures based on WordNet [28, 29, 30].

Distributional measures of word-distance rely on a **distributional hypothesis**, which states that words that occur in similar contexts tend to be semantically close [31, 32]. Many distributional approaches represent the sets of contexts of the target words as points in multidimensional co-occurrence space. Different metrics (e.g., cosine similarity, α-skew divergence [33]) can be used to measure distributional distance between two words.

In this context, IoT scenarios are characterized by a high heterogeneity of data representation. Additionally, creating and maintaining lexical databases have proven to be time consuming tasks that requires the involvement of linguistic experts. The combination of these factors is considered to be a major drawback for evaluating semantic distance based on lexical resources in IoT scenarios. Furthermore, there is usually a lag between the current state of language usage/comprehension and the lexical resource representing it.

On the other hand, methods based on distributional profile do not require a lexical database. However, these methods require a large corpus which is consider to be a disadvantage in IoT scenarios, where the associated vocabulary is generally poor and the corpus extracted from the information shared by IoT devices is not suitable to learn distributional profiles. Creating and maintaining a large corpus for IoT scenarios, as in the case of lexical databases, are time consuming tasks that requires the intervention of domain experts.

In [12], authors study the application of semantic methods for M2M scenarios and proposed the use of external public services (e.g., conventional search engines) as a replacement for large corpus, and as a solution to the rather poor vocabulary associated with M2M scenarios. In the current paper we will leverage these concepts for the implementation of a flexible IoT service discovery mechanism in the context of ICN.

2.3. Service Discovery for IoT environments

Although discovery is a well-studied subject and a mature technology in traditional networks, efficient service discovery for the IoT remains a challenge. IoT environments are generally highly dynamic (e.g., physical mobility, radio duty cycles, low power and lossy environments) and involve a massive amount of heterogeneous (e.g., disparate communication and computation resources, structure for sharing information) nodes targeted by different applications. These characteristics raise different issues for an effective and efficient discovery (e.g., availability, scalability, interoperability), which consequently require a high degree of automation (e.g., self-configuring, self-managing, self-optimizing).

Centralized solutions ease the management of service registries, ensuring their consistency and providing fast lookup mechanisms. However, relying in decentralized solutions and allowing the proactive advertisement of services are key elements for increasing the solution scalability for IoT environments. In order to make information useful and to ensure interoperability among the heterogeneity of devices and applications, it is necessary to provide a meaningful description of the services (e.g, functionality, scope, behaviour, QoS) as well as a flexible matchmaking (e.g., use of semantical information). Due to the pervasive nature and the sensibility of information commonly associated to IoT scenarios and applications (e.g., smart healthcare, logistics, transportation), handling security and privacy are other major challenges associated to IoT discovery solutions. Additionally, discovery systems should account for constant changes in the topology, keeping the information updated and ensuring load-balancing and fault tolerance.

Authors in [34] provide a comprehensive survey on service discovery approaches and define the prime criteria that need to be fulfilled for an autonomic service discovery. Screened solutions were 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 provisioning of semantic service description and capabilities is identified as a key element for service discovery automation.

Recent research on discovery solutions for IoT environments has been focusing on the different challenges we have previously identified at the beginning of the section. In [35], authors propose a Service Discovery solution which relies on ZeroConf mechanisms and P2P technologies for integrating discovery mechanisms in both local and large scale. A fully distributed opportunistic approach is used in [36] to optimise the discovery of services offered by constrained nodes. The proposed solution leverages the broadcast nature of the wireless channel to optimise discovery tasks and discovery message are transmitted using link-layer broadcasts to all neighbours which will cooperatively make the next decision.

Other approaches have proposed the use of semantic features/methods as a key element for supporting interoperability among the heterogeneous entities composing the IoT. In [37], authors point out that most work related with IoT interoperability has mostly focused on resource management, and not on how to utilize the information generated. They proposed a description ontology for the IoT Domain by integrating and extending existing work in modelling concepts in IoT. In [38], a semantic-based IoT service discovery system is proposed. The solution is distributed over a hierarchy of semantic gateways and relies on dynamic clustering of discovery information. This work is further extended in [39] with new mechanisms to handle service mobility in order to account for dynamic environments. A unified semantic knowledge base for IoT is presented in [40], consisting of several ontologies, namely resources, services, location, context, domain and policy. Semantic modelling is also considered in [41], which introduces an IoT component model and based on that model proposes an IoT directory that supports semantic description, discovery and integration of IoT objects.

The previous solutions mostly rely on ontologies to organize and discover information in IoT scenarios. Each work defines a new ontology or extends an existing one to better suit specific scenarios. However, as explained in [42, 43, 44], the use of ontologies requires the definition of entities and their relations *a priori*. Consequently, this approach hinders the compatibility between platforms and limits the quantity of information that can be shared/used in IoT environments, thus constraining their future developments.

Other works [45, 46] share our motivation and propose a vocabulary free approach for an approximate semantic matching of events to tackle the challenges (e.g., schema maintenance, model agreement) associated to the semantic heterogeneity of IoT environments. However, their work focuses on event publishing and matching, relying in thesaurus and Wordnet to define a semantic metric. As pointed out in section 2.2 concept-distance metrics that rely in lexical resources are not ideal for IoT scenarios. Our work focuses instead in the semantic features that can be used in generic IoT scenarios.

In the current work we focus on enabling semantic matchmaking of services, ensuring high reasoning levels. Other aspects of the service discovery process, such as exploring different levels of centralization will be addressed in future stages of this work.

3. Problem statement

The IoT is expected to comprise a plethora of heterogeneous devices with different ways of describing the information they produce. This fact hinders the interoperability among different applications, which although desiring/providing information with similar context use different vocabulary. In this context, the evaluation of the semantic similarity of different concepts appears as a promising area in breaking the resulting informational silos. The use of semantic similarity mechanisms could provide a decisive contribution towards the exploration of ICN architectures in IoT environments. Namely, the application of matching mechanisms into the content reaching operations of the networking fabric itself can be used to have a network that better mimics the complex relationships between devices (e.g., sensors, actuators), their generated content (e.g., temperature values with different units) and its dissemination towards interested entities.

As such, our main target in the current paper is to explore inference mechanisms at the application layer of ICN, specifically for the implementation of a broker-based service discovery mechanism with flexible query/service matching capabilities.

4. Solution overview

The current section introduces the main concepts, entities and communication procedures related to our solution.

4.1. Solution Description

Our solution considers, as shown in Figure 1, four basic entities: (i) Clients, (ii) Service Providers, (iii) Discovery Brokers and (iv) Semantic Matching Engines (SME). The different entities interact with each other through the use of well defined interfaces and their principal functions may be described as follows:

1) *Client:* An entity interested in a certain information (e.g., actuators, end user terminals). It communicates, using the NDN protocol, with the Discovery Broker through the interface Ic and with the Service Providers through the interface Ir. Clients support two operations: *(i)* Service Discovery: The client issues a request to the Discovery Broker to find out the available services which are providing content suitable to its needs; *(ii)* Content Retrieval: The client issues a content request to a given Service Provider, which in turn provides it with the desired piece of content.
2) *Service Provider:* An entity providing one or more services (e.g., sensors, actuators). It communicates, using the NDN protocol, with the Discovery Broker through the interface Is and with the interested Clients through the interface Ir. Service Providers, support two operations: *(i)* Service (Un)Registering: Sends a request to the Discovery Broker in order to add/remove its services to/from the list of services it announces to potential clients; *(ii)* Content Providing: Listens/Satisfies interests from potential clients and provides them with the corresponding content.
3) *Discovery Broker:* The entity responsible for holding the information about the available services and for matching incoming queries against the available services (by interacting with the Semantic Matching Engine). It communicates, using the NDN protocol, with the interested Clients through the interface Ic and with the Service Providers through the interface Is. It also communicates with the SME over an available transport protocol (e.g., UDP, TCP, ICN) through the interface Im. In this work, the SME is considered to be an external entity with respect to the Discovery Broker, able to be interfaced by appropriate mechanisms. This allows, for example, the possibility of

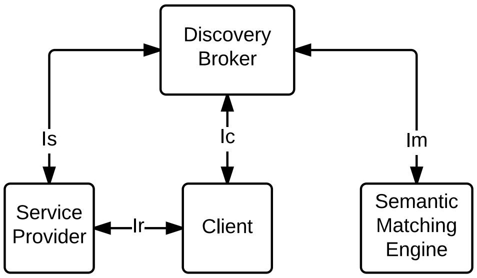


Figure 1: Solution overview: entities and interfaces

accommodating different kinds of semantic engines simultaneously. Nonetheless, the framework is flexible enough to consider the SME as an intrinsic part of the Discovery Broker if such an approach simplifies or favours the deployment of the solution (e.g., by using transport over UNIX_SOCKET). However, for the purpose of this paper, we have focused on the matching capabilities provided by the SME. The functions of the Discovery Broker include: *(i)* Service (Un)Registering: Listens for requests from potential Service Providers, and accordingly adds/removes services to/from the local table of available services and forwards part of the received information to the Semantic Matching Engine in order to keep updated the services database located at the matching engine; *(ii)* Service Matching: Listen for discovery queries from clients, forwards them to the Semantic Matching Engine and based on its response, answers to the client with a list of the matching services.

4) *Semantic Matching Engine:* The entity responsible for performing the actual matching of queries and services. It keeps track of the registered services, and matches the incoming queries with the available services. It communicates, over an available transport protocol, with the Discovery Broker through the interface Im. It has two main functions: *(i)* Service (Un)Registering: Listens for requests coming from the Discovery Broker and accordingly adds/removes services form its local table and give the relevant feedback to the broker; *(ii)* Service Matching: Listens for queries coming from the Discovery Broker, runs the different matching algorithms and replies with a list of the relevant services (i.e. services for which there is a positive matching between the terms included in the query and the tags used to describe the service).

4.2. Semantic Matching Engine: Detailed Description

In the current paper we extend the core concepts of the solution proposed in [12] with novel functionalities for supporting service discovery mechanisms turning it into a full fledged Semantic Matching Engine. Added functionalities include (un)registration of services, process incoming service discovery queries, match query terms with service description tags, respond with the results of the matchmaking process.

The solution relies on web search engines to extract the distributional profiles of words (i.e., the weighted neighbourhood of the word). The resulting system, as depicted in Figure 2, receives two terms as input and returns the semantic similarity between them. Cosine similarity (Equation (1)) is used to evaluate the proximity between the two terms. Distributional profiles are either available at the local cache or need to be otherwise extracted. The process of calculation of the distributional profiles comprises three major components *(i)* Corpus Extraction, which acts as a bridge between the solution and the search engine (i.e., Bing[4] and Faroo[5] APIs); *(ii)* Text Processing, a pipeline that process and cleans the corpus; *(iii)* Distributional Profile Extraction, which analyses the output of the previous pipeline and extract the profile of the term. The initial work in [12] extracted distributional profiles based only in unigrams, while here we handle unigrams, bigrams and trigrams. Additionally, a filtering mechanism for removing low frequency dimensions and consequently improving system accuracy was introduced. This mechanism is based on the elbow method, which is commonly used to select the ideal number of clusters for a given population.

The Semantic Matching Engine, besides the described semantic similarity mechanism, also provides matching information based on exact string matching (i.e., returns 1 or 0 depending on whether the words are the same or not) and matching within a certain Levenshtein distance (i.e. a given number of single-character edits). For comparing the similarity of set of words Jaccard Index (Equation (2)) and Cosine similarity are considered.

$$\cos(A,B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

4.3. Detailed Communication Procedures

This subsection presents a detailed description of the procedures followed by the different entities to communicate with each other.

4.3.1. Service (Un)Registration Procedure

Services, in order to be discoverable, must register on the Discovery Broker as shown in Figure 3. A Service Provider, sends a registration interest, $Interest(1)$, to the broker responsible for its namespace. The registration contains relevant information about the service(s) being registered (e.g., unique id, name, metadata, semantic description). The broker registers the service(s) and sends back $Data(2)$ to the Service Provider with the result of the operation which in case of

[4] `www.bing.com`

[5] `www.faroo.com`

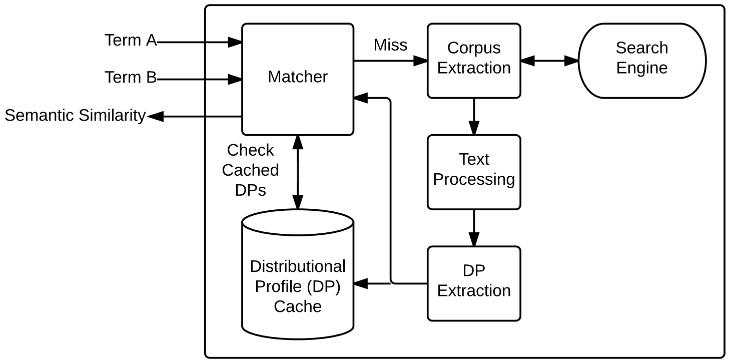


Figure 2: Semantic Matching Procedure

collision with already registered services (i.e., id or name) provides alternative values for the colliding parameters. Once the Broker has registered the services it sends, $Request(3)$, with the semantic description of the services to the Semantic Matcher and receives back the results of the operation, $Response(4)$. The service unregistration process follows a similar procedure, $Packets(5-8)$, however only the ids of the services are included in the unregistration requests.

4.3.2. Service Discovery Procedure

Clients, as shown in Figure 4, in order to discover the available services must send a query, $Interest(1)$, to the Discovery Broker. The query includes a semantic description of the desired services. The broker forwards the request to the Semantic Matcher, $Request(2)$, which determines the set of relevant services and returns the corresponding ids to the broker, $Response(3)$. The broker processes these ids and returns the full description of the services back to the client, $Data(4)$. Afterwards, the client can directly request the content to the Service Providers according to the principles of the ICN architecture being used.

5. Evaluation

In this section we evaluate our proposal by deploying a proof-of-concept prototype into an experimental environment. In validating our proposal, we focused on three 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), *(ii)* the overhead introduced in the network and *(iii)* the performance of different matching algorithms.

5.1. Proof-of-concept prototype

For implementing the proof-of-concept prototype we selected the NDN architecture and based its development on the NDN C++ library with eXperimental

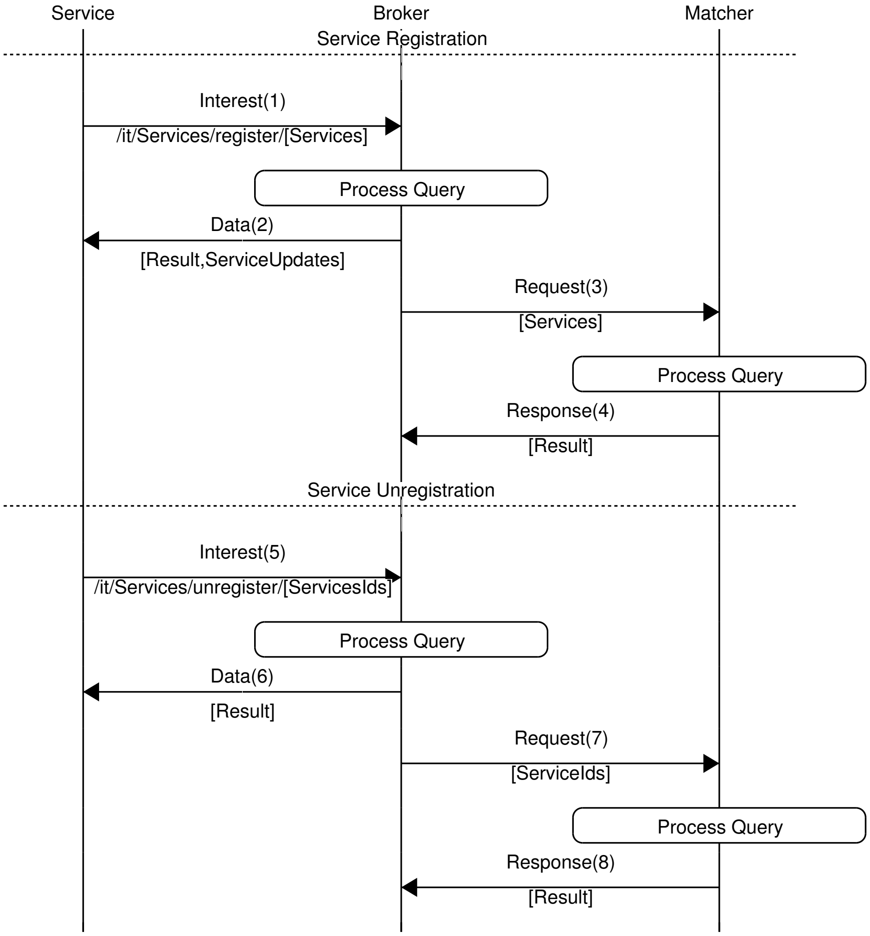


Figure 3: Service (un)registration message sequence

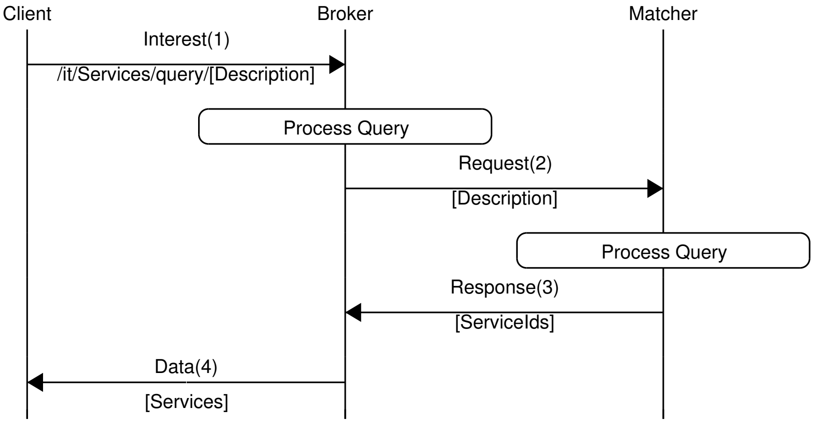


Figure 4: Service discovery message sequence

eXtensions (ndn-cxx) and NDN Forwarding Daemon (NFD) implementations (version 0.3.2)[6]. The semantic matcher was implemented in Java and the communication between the matcher and the broker was performed over UDP. The information exchanged using NDN was encoded using TLV, while the information exchange over UDP was encoded using JSON.

5.2. Evaluation environments

For the evaluation of our implementation we deployed the prototype in an experimental testbed. The semantic matcher was deployed in a virtual machine (single core 3.33GHz virtualised CPU with 2GB of RAM) hosted in an OpenStack Platform and connected through Gigabit Ethernet. The remaining entities were deployed in separate nodes of the AMazING testbed [47]. Each node runs an Ubuntu 12.04 OS on top of a hardware configured with a VIA Eden 1GHz processor with 1GB RAM, a 802.11a/b/g/n Atheros 9K wireless interface, and a Gigabit wired interface. For our evaluation, we deployed our solution in a simple scenario composed by a Broker, a Semantic Matcher, a single Client and a single Server. The evaluation scenario has as main goals to assess the feasibility of the proposed solution and to identify of its main challenges, not focusing on scalability aspects.

5.3. Evaluation Dataset

A key element for the evaluation of the performance of the developed prototype is the use of a representative dataset. By analysing the applications offered by IoT Platform Providers (e.g., libelium[7], carriots[8]) we extracted a

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

[7] `http://www.libelium.com`

[8] `https://www.carriots.com`

Table 1: Groups of Query

Group	Description	Sample Terms
M2M	Exact Match	moisture, greenhouse, soil, agriculture
E2M(1/1)	One word with one error	moisture**s**, greenhouse, soil, agriculture
E2M(1/2)	One word with two errors	moistur**is**, greenhouse, soil, agriculture
E2M(2/2)	Two words with one error each	moisture**s**, greenhouse**s**, soil, agriculture
U2M(1)	One word replacement	**wetness**, greenhouse, soil, agriculture
U2M(2)	Two words replacement	**wetness**, **hothouse**, soil, agriculture
U2M(3)	Three words replacement	**wetness**, **hothouse**, **ground**, agriculture
U2M(4)	Four words replacement	**wetness**, **hothouse**, **ground**, **cultivation**

set of terms commonly associated to IoT services as well as different ways of referring to them. Using this information we designed a dataset that properly describes scenarios expected to be part of the IoT (e.g., Smart Cities, Smart Agriculture, Domotic, Home Automation). The dataset is composed of services and queries each of which is described by 4 keywords. In the case of the queries we considered 3 different approaches: *(i)* Machine-to-Machine (M2M) scenarios – the requester knows the exact keywords that better represent the service, *(ii)* Engineer-to-Machine (E2M) – the requester has the knowledge of the proper keywords, but is subjected to typing mistakes, *(iii)* User-to-Machine (U2M) – the requester has some knowledge about the service but does not know the exact keywords so it would most likely use synonyms of proper keywords. Following these approaches, and varying the number of errors/synonyms included in the query, we defined 8 groups of queries as described in Table 1. The resulting dataset is composed by 30 services and 240 queries. Each service has 8 queries associated, each of which falls into one of the mentioned groups.

5.4. Solution performance evaluations

The current section describes the conducted evaluations and presents the obtained results.

5.4.1. Service Time

We evaluated the service time for the three main operations of our solution: register service, unregister service and service query (see Figures 3 and 4). The number of services being processed in each evaluation varied from 1 to 30 (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 studied: *(i)* all services in a single aggregated request (*all-at-once*), and *(ii)* one service per request. This last approach was divided into two sub-approaches depending on whether the requester waits (*one-by-one*) or not (*one-at-once*) for an answer before sending the next request. In the case of one service per request, the amount of time considered is the total time elapsed from the moment the first request is sent, until the reception of the last response. All evaluations were run 10 times and a 95% confidence interval was calculated.

The results of these assessments are presented in Figure 5. Figure 5a shows the service time for the service discovery operation performed by the Clients[9]. As expected, the discovery time and the number of registered services exhibit a direct relation, not only because of the increase of the reply size but also due to the increase of the processing time at the semantic matcher.

Figures 5b and 5c show the results for the registration and unregistration process respectively. Results show that the service time for unregistration procedures are smaller than those from the registration procedures, mainly due to the fact that while registration requests involve a full description of the service, unregistration request involves only the a numeric identifier of the service. Using the *all-at-once* approach, results show that there is not a 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-by-one* and *one-at-once* approaches resulted in a significant increase of the service time. The reason behind this behaviour includes the involvements of larger network overhead (as will be seen in the next subsection) and also due to the need of processing a larger amount of packets at the different layers of the network stack.

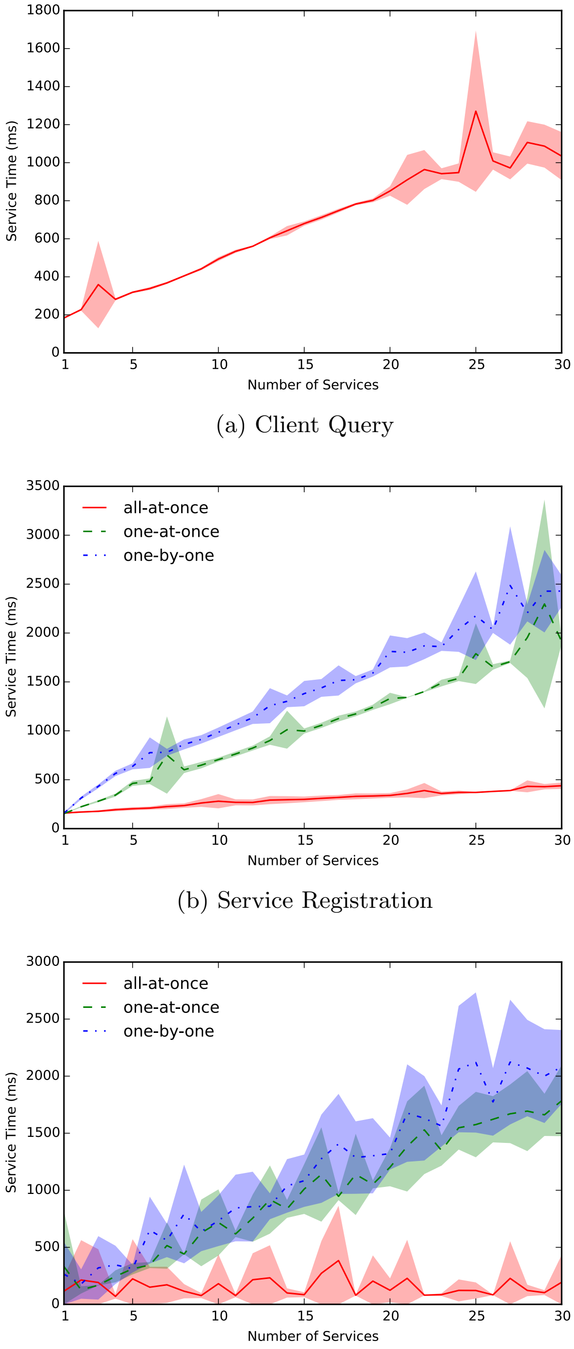
5.4.2. Network Overhead

This subsection provides an analysis of the network overhead at each interface of our solution. Table 2 shows the results for our main scenario involving 30 services and for the two approaches studied in the previous section (i.e., services (un)registration requests are sent on individual packets or aggregated in a single packet). As expected, the larger overhead is associated to the interface Is. Consequently, the aggregation of services in the same request leads to a significant reduction of the network overhead, particularly for the interfaces Is and Im, the overhead for the interfaces Ic and Ir is not affected by the approach used for (un)registering the services. The overhead associated with a single content request over the interface Ir (actual content retrieval) represents a 0.96% and 3,63% of the overhead associated to the service discovery process for the individual request and aggregated request strategies respectively. However, it is typical that after discovering a service the client will interact with the service provider several times and as the number of requests augments the service discovery overhead will be less significant.

5.4.3. Semantic matching performance

We evaluated the performance of the different string matching algorithms (i.e., exact string matching, Levenshtein distance of 2 and semantic similarity) over the whole evaluation dataset, using two different statistics for comparing the similarity of the set of words (i.e., Jaccard Index and Cosine similarity). However, for all the cases the results obtained for Jaccard and Cosine were

[9] The results only show the behaviour for one of the evaluation cases as the way services are (un)registered does not affect the time taken by the discovery process



(a) Client Query

(b) Service Registration

(c) Service Unregistration

Figure 5: Service Time

Table 2: Network Overhead

	Network Overhead (bytes)	
Interface	Individual Request	Aggregated Request
Is	36988	7538
Ic	3623	3623
Im	12359	2919
Ir	511	511

almost identical and therefore for the remaining of this subsection we will be presenting only the results obtained for the Cosine similarity.

Figure 6 represents the average precision of the answers provided by each of the string matching algorithms. In the figure the small squares represent a query (e.g., the query within the group "M2M" that is associated with service "0") while its colour tone indicates the obtained average precision. In calculating the average precision we used Equation (3), where k is the rank in the sequence of retrieved documents, n is the number of retrieved documents, $P(k)$ is the precision (i.e., the fraction of the retrieved documents that are relevant) at cut-off k in the list and $rel(k)$ is an indicator function equal to 1 if the item at rank k is a relevant document and zero otherwise. For our evaluations, we considered as relevant only the service associated with the query.

$$AP = \frac{\sum_{i=1}^{n}(P(k) \times rel(k))}{number\ of\ relevant\ documents} \tag{3}$$

Figure 7 represent the Mean Average Precision values in a form of a boxplot where the lines represent the 95% confidence interval for the results. Using the same representation scheme.

From figures Figure 6 and 7 it can be observed that exact string matching and Levenshtein distance present a great precision for the first groups, but queries with more than 2 synonyms are not properly match to the relevant service. However the semantic similarity matching still manages to get the matching service, although not in the proper rank.

From Figure 8, which represents the processing time for the different matching algorithms, it can be established that the semantic matching is a time consuming process, thus introducing delay in the service discovery process and therefore requiring further attention.

An analysis of these results (Figures 6–8) show that the current approach constitute a first step into further refinements of the semantic matching algorithm. However, they demonstrate the feasibility of using such techniques. Particularly for the case of the queries that include 3 and 4 synonyms, where the conventional methods did not obtain a match for the service, but the semantic method was able to find some matches. The results also point out as future strategies to consider not only the individual results for each of the mechanisms, but also a weighted sum of these individuals results. The low performance of the semantic mechanism on the E2M groups suggests the possibility of considering

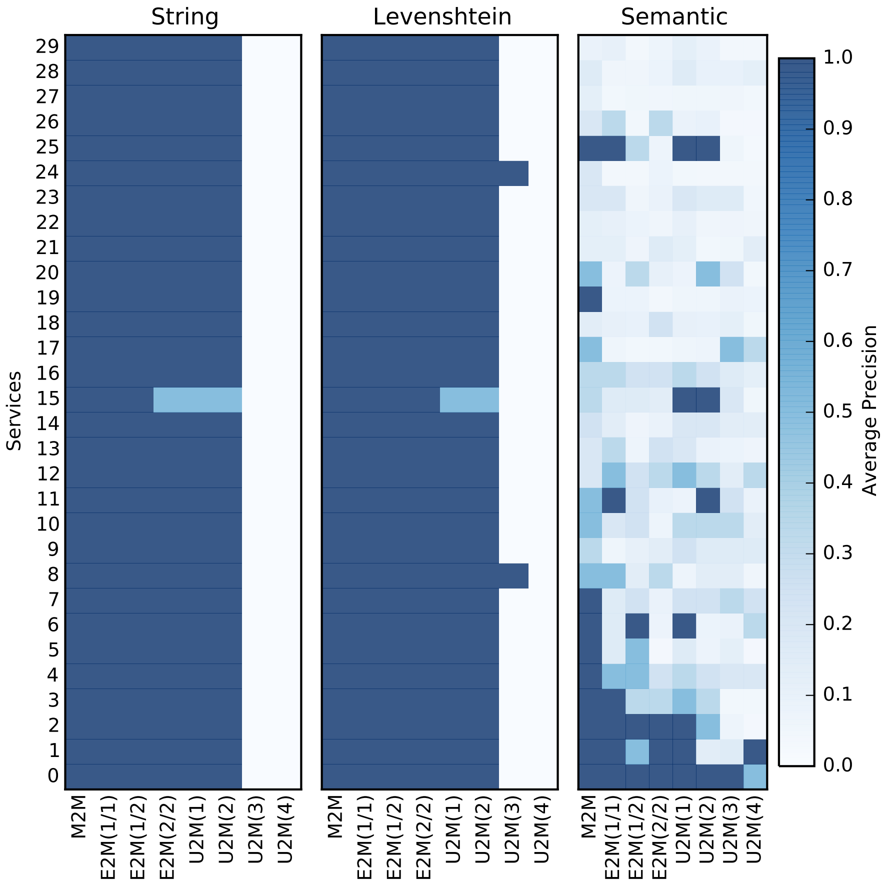


Figure 6: Average Precision Heatmap

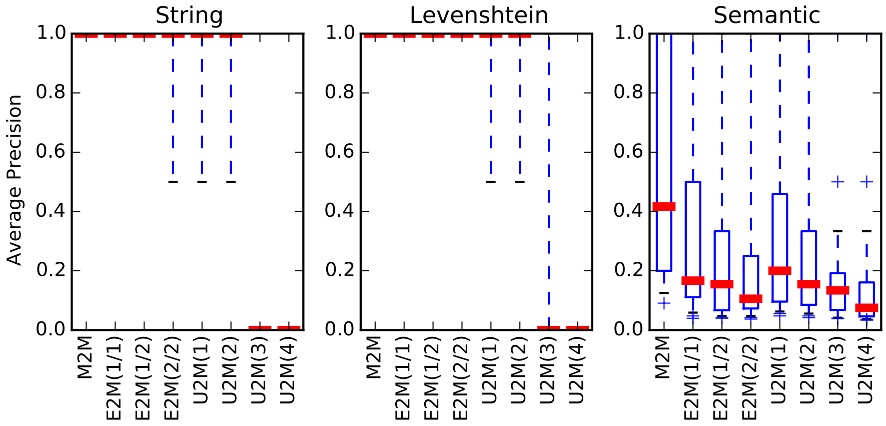


Figure 7: Mean Average Precision Boxplot

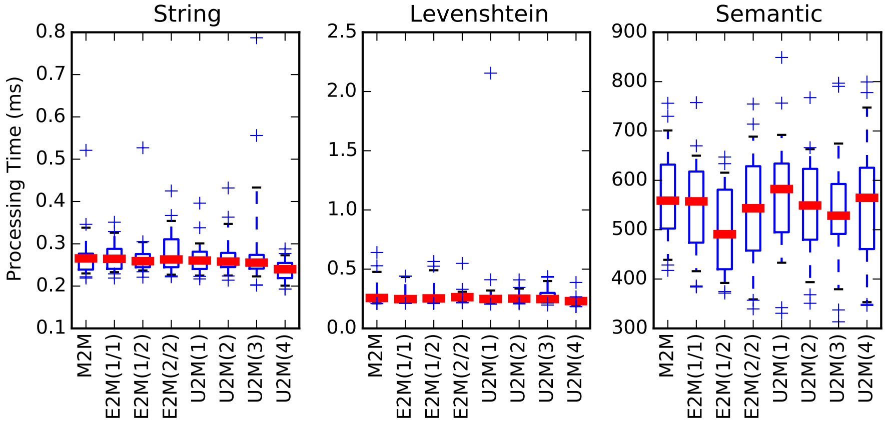


Figure 8: Processing Time Boxplot

words within the Levenshtein distance during the evaluation of the distributional profiles of a given term. The use of words thesaurus may also be leveraged for an improved performance. A second issue requiring further attention is the relatively high processing time of the semantic matching mechanism. A possible way of addressing this issue is to extend the cache not only to the extracted corpus, but also to the results of distributional profile comparisons.

6. Conclusions

In this paper we showcased the possibilities that arise from the application of Semantic Matching to the Information Centric Networking, more specifically to Service Discovery in Interest-based ICN. As a proof of concept for this approach, a prototype of a discovery protocol was developed and tested experimentally. Results show that although further improvements are required, the use of a semantic matcher as part of the service discovery solution increases its flexibility allowing the correct matching of queries and services where none of the words are an exact match but synonyms instead.

Additionally, it is important to highlight that the application of the semantic matching concepts into ICN scenarios should not be limited to those presented in the current paper and, in future works, we plan to extend the application of matching engines to the network layer itself (e.g, forwarding in meaningful namespaces, routing in flat namespaces). Also, future deployments of this solution may explore alternative software, specifically targeting IoT devices, such as RIOT OS[10] [48], which is an operating system for IoT devices, and CCN-Lite[11], a lightweight solution compliant with different Interest-based ICN implementations.

Acknowledgements

This work was supported by project Cloud Thinking (CENTRO-07-ST24-FEDER-002031), co-funded by QREN, Mais Centro programme. In 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. And also under the research grant SFRH/BD/94270/2013.

References

[1] D. Miorandi, S. Sicari, F. D. Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks 10 (7) (2012) 1497 – 1516.

[10] http://www.riot-os.org/

[11] http://www.ccn-lite.net/

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (7) (2013) 1645 – 1660. doi:http://dx.doi.org/10.1016/j.future.2013.01.010.

[3] CISCO, Cisco visual networking index: Global IP traffic forecast, 2014 - 2019, Tech. rep., CISCO (2015).

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, Communications Magazine, IEEE 50 (7) (2012) 26–36.

[5] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, A survey of information-centric networking research, Communications Surveys Tutorials, IEEE 16 (2) (2014) 1024–1049.

[6] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies, A. Molinaro, S. Eum, Information-Centric Networking: Baseline Scenarios, RFC 7476 (Informational) (Mar. 2015).

[7] A. Lindgren, F. B. Abdesslem, B. Ahlgren, O. Schelen, A. Malik, Applicability and tradeoffs of information-centric networking for efficient iot, in: Internet-Draft, 2015.

[8] Y. Zhang, D. Raychadhuri, L. Grieco, E. Baccelli, J. Burke, R. Ravindran, G. Wang, Icn based architecture for iot - requirements and challenges, in: Internet-Draft, 2015.

[9] M. Antunes, D. Gomes, R. Aguiar, Context storage for m2m scenarios, in: Communications (ICC), 2014 IEEE International Conference on, 2014, pp. 3664–3669.

[10] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone, L. Veltri, A scalable and self-configuring architecture for service discovery in the internet of things, Internet of Things Journal, IEEE 1 (5) (2014) 508–521.

[11] M. Antunes, D. Gomes, R. L. Aguiar, Scalable semantic aware context storage, Future Generation Computer Systems 56 (2016) 675 – 683. doi:http://dx.doi.org/10.1016/j.future.2015.09.008.

[12] M. Antunes, D. Gomes, R. Aguiar, Semantic features for context organization, in: Future Internet of Things and Cloud (FiCloud), 2015 International Conference on, 2015.

[13] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al., Named data networking, ACM SIGCOMM Computer Communication Review 44 (3) (2014) 66–73.

[14] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking named content, in: Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT '09, ACM, 2009, pp. 1–12.

[15] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, A. V. Vasilakos, Information-centric networking for the internet of things: Challenges and opportunities, IEEE Network Magazine.

[16] N.-T. Dinh, 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.

[17] Z. Ren, M. Hail, H. Hellbruck, 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, 2013, pp. 123–128. `doi:10.1109/ISSNIP.2013.6529776`.

[18] J. Burke, P. Gasti, N. Nathan, G. Tsudik, Securing Instrumented Environments over Content-Centric Networking: the Case of Lighting Control and NDN, in: IEEE NOMEN Workshop, 2013.

[19] J. Burke, et al., Secure Sensing over Named Data Networking, in: IEEE Network Computing and Applications (NCA), 2014, pp. 175–180.

[20] M. Amadeo, C. Campolo, A. Molinaro, Multi-source data retrieval in iot via named data networking, in: Proceedings of the 1st International Conference on Information-centric Networking, INC '14, ACM, New York, NY, USA, 2014, pp. 67–76.

[21] D. Corujo, R. L. Aguiar, I. Vidal, J. García-Reinoso, K. Pentikousis, 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, 2014, pp. 1–5.

[22] J. Quevedo, D. Corujo, R. Aguiar, A case for icn usage in iot environments, in: Global Communications Conference (GLOBECOM), 2014 IEEE, 2014, pp. 2770–2775.

[23] J. Quevedo, D. Corujo, R. Aguiar, Consumer driven information freshness approach for content centric networking, in: Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, 2014, pp. 482–487. `doi:10.1109/INFCOMW.2014.6849279`.

[24] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, M. Wählisch, Information Centric Networking in the IoT: Experiments with NDN in the Wild, in: 1st ACM Conference on Information-Centric Networking (ICN-2014), 2014, pp. 77–86. `arXiv:1406.6608`, `doi:10.1145/2660129.2660144`.

[25] G. Scott, Ccnx 1.0 simple service discovery, Tech. rep., Computing Science Laboratory, Palo Alto Research Center (2014).

[26] R. Ravindran, T. Biswas, X. Zhang, A. Chakraborti, G. Wang, Information-centric networking based homenet, in: Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, 2013, pp. 1102–1108.

[27] G. A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (11) (1995) 39–41.

[28] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics, ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA, 1994, pp. 133–138.

[29] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95, 1995.

[30] S. Banerjee, T. Pedersen, An adapted lesk algorithm for word sense disambiguation using wordnet, in: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, CICLing '02, 2002, pp. 136–145.

[31] J. Firth, A synopsis of linguistic theory 1935-55, 1957.

[32] Z. S. Harris, Mathematical structures of language, Wiley, 1968.

[33] L. Lee, On the effectiveness of the skew divergence for statistical language analysis, in: Artificial Intelligence and Statistics, 2001, pp. 65–72.

[34] M. Rambold, H. Kasinger, F. Lautenbacher, B. Bauer, Towards Autonomic Service Discovery A Survey and Comparison, in: 2009 IEEE International Conference on Services Computing, no. Section II, Ieee, 2009, pp. 192–201. doi:10.1109/SCC.2009.59.

[35] S. Cirani, L. Davoli, G. Ferrari, A scalable and self-configuring architecture for service discovery in the internet of things, Internet of Things ... 1 (5) (2014) 508–521.

[36] B. Djamaa, M. Richardson, P. Barker, I. Owens, Discovery of Things: A Fully-Distributed Opportunistic Approach, 2015 IEEE 81st Vehicular Technology Conference (VTC Spring) (2015) 1–5doi:10.1109/VTCSpring.2015.7145778.

[37] W. Wang, S. De, R. Toenjes, E. Reetz, K. Moessner, A comprehensive ontology for knowledge representation in the internet of things, in: Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on

Ubiquitous Computing and Communications, IUCC-2012, 2012, pp. 1793–1798. doi:10.1109/TrustCom.2012.20.

[38] S. B. Fredj, M. Boussard, D. Kofman, L. Noirie, A scalable IoT service search based on clustering and aggregation, in: Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013, 2013, pp. 403–410. doi:10.1109/GreenCom-iThings-CPSCom.2013.86.

[39] S. B. Fredj, M. Boussard, Efficient semantic-based IoT service discovery mechanism for dynamic environments, Personal, Indoor, and ... (2014) 2088–2092.

[40] S. Nambi, C. Sarkar, R. V. Prasad, A. Rahim, A unified semantic knowledge base for IoT, in: Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, pp. 575–580. doi:10.1109/WF-IoT.2014.6803232.

[41] S. Chun, S. Seo, B. Oh, K.-H. Lee, Semantic description, discovery and integration for the Internet of Things, in: Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), 2015, pp. 272–275. doi:10.1109/ICOSC.2015.7050819.

[42] C. Shirky, Ontology is overrated: Categories, links, and tags, http://shirky.com/writings/ontology_overrated.html, accessed: December 2015 (May 2005).

[43] G. Avram, At the crossroads of knowledge management and social software, Electronic Journal of Knowledge Management 4 (1) (2006) 1–10.

[44] T. Gruber, Ontology of folksonomy: A mash-up of apples and oranges, International Journal on Semantic Web and Information Systems 3 (2) (2007) 1–11.

[45] S. Hasan, E. Curry, Approximate Semantic Matching of Events for the Internet of Things, ACM Transactions on Internet Technology 14 (1) (2014) 1–23. doi:10.1145/2633684.

[46] S. Hasan, S. O'Riain, E. Curry, Approximate semantic matching of heterogeneous events, in: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems - DEBS '12, ACM Press, New York, New York, USA, 2012, pp. 252–263. doi:10.1145/2335484.2335512.

[47] J. Barraca, D. Gomes, R. Aguiar, Amazing - advanced mobile wireless playground, in: T. Magedanz, A. Gavras, N. Thanh, J. Chase (Eds.), Testbeds and Research Infrastructures. Development of Networks and Communities, Vol. 46 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, 2011, pp. 219–230.

[48] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, T. Schmidt, Riot os: Towards an os for the internet of things, in: Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, 2013, pp. 79–80. doi:10.1109/INFCOMW.2013.6970748.