Universidade de Aveiro Departamento de Eletrónica,
Telecomunicações e Informática
2020

José Rafael
Quevedo Rego

Incrementando as Redes Centradas à Informação para uma Internet das Coisas baseada em Nomes

Enhancing Information-Centric Networking for a name-based Internet of Things

Universidade de Aveiro Departamento de Eletrónica,
Telecomunicações e Informática
2020

José Rafael Quevedo Rego

Incrementando as Redes Centradas à Informação para uma Internet das Coisas baseada em Nomes

Enhancing Information-Centric Networking for a name-based Internet of Things

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2020

José Rafael Quevedo Rego

Incrementando as Redes Centradas à Informação para uma Internet das Coisas baseada em Nomes

Enhancing Information-Centric Networking for a name-based Internet of Things

Tese apresentada às Universidades de Minho, Aveiro e Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor no âmbito do programa doutoral MAP-tele, realizada sob a orientação científica do Doutor Daniel Nunes Corujo, Investigador Doutorado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Rui Luís Andrade Aguiar, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Este doutoramento foi realizado com o apoio do Instituto de Telecomunicações - Pólo de Aveiro (UID/EEA/50008/2019)

To my parents, they gave me everything, and sacrifice even more for me to reach this point.

o júri / the jury

presidente / president	Doutor Helmuth Robert Malonek Professor Catedrático, Universidade de Aveiro
vogais / examiners committee	Doutor Paulo Alexandre Ferreira Simões Professor Auxiliar, Universidade de Coimbra
	Doutor Fernando Manuel Valente Ramos Professor Auxiliar, Universidade de Lisboa
	Doutor Joaquim Arnaldo Carvalho Martins Professor Catedrático, Universidade de Aveiro
	Doutor Aníbal João De Sousa Ferreira Professor Associado, Universidade do Porto
	Doutor Daniel Nunes Corujo Investigador Doutorado, Universidade de Aveiro

agradecimentos /
acknowledgements

I would like to start by thanking my supervisors, Dr. Daniel Corujo and Prof. Dr. Rui. L. Aguiar, not only for providing guidance and support during the realization of this work but also for contributing to my personal and professional growth.

I also thank the Instituto de Telecomunicações - Pólo Aveiro (ITAV) for providing support for conducting this research. In particular, I am thankful to the Telecommunications and Networking - Av group from the referred institution.

This thesis brings to an end many hours of research, where I had the opportunity to collaborate with many colleagues, with whom I had thoughtful discussions of various problems which definitively contributed to improving the quality of this work. To all of them, my most sincere appreciation.

Finally, my deepest gratitude to my family and friends who have been an integral part of this journey.

Palavras Chave Internet do Futuro; Redes Centradas à Informação; Internet das Coisas;

Resumo

A forma como usamos a Internet tem vindo a evoluir desde a sua criação. Atualmente, os utilizadores estão mais interessados em aceder a conteúdos e serviços, com elevados requisitos em termos de largura de banda, segurança e mobilidade. Esta evolução desencadeou o desenvolvimento de novas arquiteturas de rede, visando os atuais, bem como os futuros, requisitos de utilização. As Redes Centradas à Informação (*Information-Centric Networking - ICN*) são um exemplo proeminente destas novas arquiteturas que, em vez de seguirem um modelo de comunicação centrado nos dispositivos terminais, centram as suas funções de rede em torno do próprio conteúdo.

Paralelamente, novos cenários de utilização onde dispositivos inteligentes interagem entre si, e com outros elementos de rede, têm vindo a aparecer e constituem o que hoje conhecemos como a Internet das Coisas (*Internet of Things - IoT*). É esperado que a IoT tenha um impacto significativo na economia e na sociedade. No entanto, promover a adoção em massa da IoT ainda requer que muitos desafios sejam superados. Apesar dos desenvolvimentos recentes, vários problemas relacionados com a adoção em larga escala de soluções de IoT baseadas no protocolo IP estão em aberto.

O facto da IoT estar focada em dados e informação, em vez de comunicações ponto-a-ponto, sugere a adoção de soluções baseadas em arquiteturas ICN. Neste sentido, este trabalho explora os conceitos base destas soluções para desenvolver uma visão completa dos principais requisitos que devem ser satisfeitos por uma solução IoT baseada na arquitetura de rede ICN. Esta visão é complementada com soluções para problemas cruciais para a adoção de uma IoT baseada em ICN. Em primeiro lugar, assegurar que a informação seja atualizada e, ao mesmo tempo, manter as vantagens do armazenamento intrínseco em elementos de rede das arquiteturas ICN. Em segundo lugar, permitir as funcionalidades de descoberta não só em domínios locais, mas também em domínios de larga-escala. Os mecanismos propostos são avaliados através de simulações e prototipagem, com os resultados a demonstrarem a viabilidade da sua adoção. Para além disso, os resultados deste trabalho contribuem para o desenvolvimento de conceitos sólidos em direção a uma verdadeira Internet das Coisas baseada em Nomes.

Keywords Future Internet; Information-Centric Networking; Internet of Things.

Abstract

The way we use the Internet has been evolving since its origins. Nowadays, users are more interested in accessing contents and services with high demands in terms of bandwidth, security and mobility. This evolution has triggered the emergence of novel networking architectures targeting current, as well as future, utilisation demands. Information-Centric Networking (ICN) is a prominent example of these novel architectures that moves away from the current host-centric communications and centres its networking functions around content.

Parallel to this, new utilisation scenarios in which smart devices interact with one another, as well as with other networked elements, have emerged to constitute what we know as the Internet of Things (IoT). IoT is expected to have a significant impact on both the economy and society. However, fostering the widespread adoption of IoT requires many challenges to be overcome. Despite recent developments, several issues concerning the deployment of IP-based IoT solutions on a large scale are still open.

The fact that IoT is focused on data and information rather than on point-to-point communications suggests the adoption of solutions relying on ICN architectures. In this context, this work explores the ground concepts of ICN to develop a comprehensive vision of the principal requirements that should be met by an IoT-oriented ICN architecture. This vision is complemented with solutions to fundamental issues for the adoption of an ICN-based IoT. First, to ensure the freshness of the information while retaining the advantages of ICN's in-network caching mechanisms. Second, to enable discovery functionalities in both local and large-scale domains. The proposed mechanisms are evaluated through both simulation and prototyping approaches, with results showcasing the feasibility of their adoption. Moreover, the outcomes of this work contribute to the development of new compelling concepts towards a full-fledged Named Network of Things.

Contents

List of Figures

List of Tables

Acronyms

AMQP	Advanced Message Queuing Protocol
AFP	Alternative Forwarding Pipeline
AFT	Alternative Forwarding Table
AP	Access Point
CCN	Content-Centric Networking
CDN	Content Delivery Network
CICN	Community ICN
CoAP	Constrained Application Protocol
CS	Content Store
DONA	Data-Oriented Network Architecture
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
FIB	Forwarding Information Base
FIXP	Future Internet eXchange Point
ICN	Information-Centric Networking
ICNRG	Information-Centric Networking Research Group
ICT	Information and Communication Technologies
IoT	Internet of Things
IP	Internet Protocol
IRTF	Internet Research Task Force
LLN	Low-Power and Lossy Network
LoWPAN	Low-Power Wireless Personal Area Networks
LPWAN	Low Power Wide Area Networks
LRU	Least Recently Used
M2M	Machine-to-Machine
MAC	Medium Access Control
MANO	Management and Orchestration
mDNS	Multicast DNS
MEC	Mobile Edge Computing
mMTC	Massive Machine-Type Communications
MQTT	Message Queuing Telemetry Transport
MTU	Maximum Transmission Unit
NB-IoT	Narrowband IoT
NDN	Named Data Networking
NetInf	Network of Information
NFD	NDN Forwarding Daemon
NFV	Network Function Virtualization
NPSN	Named Publish Subscribe Networking
P2P	Peer to Peer
PIT	Pending Interest Table
PURSUIT	Publish-Subscribe Internet Technology
REST	REpresentational State Transfer
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
SDN	Software-Defined Networking
SLA	Service-Level Agreement
TLV	Type-Length-Value
V2V	Vehicle to Vehicle
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Networks
WSAN	Wireless Sensor and Actuator Networks

CHAPTER 1

Introduction

"We keep moving forward, opening new doors, and doing new things, because we're curious and curiosity keeps leading us down new paths."

— Walt Disney

This chapter is the first contact of the reader with the thesis document, and, as such, it starts by providing some background on the two key technologies involved in the work being presented, namely Information-Centric Networking (ICN) and Internet of Things (IoT). This background will be leveraged for a better understanding of the motivations in pursuing ICN as a networking infrastructure for the IoT in what will be denoted as ***"Named Internet of Things"****. The research framework that guided the development of this work is also presented alongside the main associated contributions. Finally, the chapter presents an overall structure of the document easing the comprehension and navigation throughout the corpus of the thesis.*

1.1 Background

The Internet is without question one of the major technological breakthroughs of recent times, and consequently, Information and Communication Technologies (ICT) are deeply rooted in a wide variety of areas in our society. Along the way, the Internet has evolved from its original purpose of interconnecting a reduced set of computers to the provision of worldwide connectivity to more than 18 billion devices [1]. This connectivity explosion, as well as the adoption of innovative applications and services, have raised several issues (e.g., scalability, security, reliability, mobility), challenging the very foundations of the Internet [2].

In this context, it is argued that the evolution of the Internet has not kept the pace of the requirements placed upon it [3]. This concern has been manifested in discussions on not only the evolution but also on radical changes to the underlying design of the Internet [4, 5]. While evolutionary approaches aim at solutions which retain backwards compatibility and could be gradually deployed, clean slate approaches propose to redesign the Internet from the ground up taking into account past mistakes, as well as current and envisioned challenges [6].

History has proven to be harsh on disruptive and not so disruptive solutions [3]. The motivation for evolutionary approaches is very straightforward, Internet Protocol (IP) works and is widespread, replacing it would involve a costly transition process. Moreover, the TCP/IP network stack has been subjected, during the last few decades, to a long process of evolution, error correction and optimisation. New approaches will require to try things out, fix arising issues and optimise functionalities.

Those who vouch for clean slate approaches claim that the emerging demands such as security, mobility, efficient content distribution are hard to be met by continuously patching up the Internet [7]. The fact is that it is *ossified*, and we are witnessing a vicious cycle in which introducing changes at the transport and network layers has never been harder [3, 8]. Consequently, the very foundations of the Internet have to be rethought. In particular, while communication in current networks is based on addressable hosts, some clean slate Future Internet architectures propose to have content itself as the main driver of the information interchange, creating an environment where content is no longer tightly coupled to its location [4].

This novel paradigm, commonly referred to as ICN [9, 10], unlike the original underlying architecture of the Internet, intrinsically supports mechanisms, such as security, mobility and efficient caching. The interest on ICN, as an enabler for a Future Internet, has been growing in the last few years, supported by industrial, academic and standardisation research efforts, illustrating the contribution that this novel design can provide in different deployments [11]. As such, it is expected that it will not only improve current network utilisations by providing more future-proof usages but also will pave the way for the emergence of a whole new set of utilisation scenarios.

In parallel, motivated by the coupling of networking communication capabilities with devices of heterogeneous characteristics (e.g., sensors, actuators), we have been witnessing the development of solutions towards an IoT [12, 13, 14]. The sensing and actuating capabilities of such devices have prompted the appearance of different smart scenarios (e.g., smart metering, smart healthcare, smart transportation), by providing a connection between the physical and digital worlds, which has generated added value and set off a trend of a continuously growing number of connected devices. According to recent studies [1], by 2022, there will be a total of 3.6 networked devices per person, where Machine-to-Machine (M2M) connections will represent 51 percent of the total devices and connections, accounting for a total of 14.6 billion connections. This connectivity explosion has placed a new set of stringent requirements over the underlying networking fabric (e.g., scalability, energy efficiency, self-organisation, semantic interoperability, privacy and security), thus creating 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 [15].

These requirements have made IoT one of the most challenging, demanding and heterogeneous areas of current ICT deployments [16, 17]. Furthermore, the suitability of the TCP/IP stack for supporting the IoT has been questioned [18], which creates an appealing opportunity for the assessment of the contribution of ICN in the provisioning of a simplified architecture towards IoT interaction. The individual operational features of ICN and the possibility of expanding its scenarios and applications, still at the design stage, have encouraged the utilisation of the ICN concepts for addressing the IoT challenges. Notably, 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 [11]. Moreover, in enabling the deployment of current and envisioned IoT scenarios over ICN architectures, the heterogeneity of requirements and scenarios, commonly associated to the IoT, are expected to provide valuable insight on the key enablers and extensions to the ICN base design. Accordingly, ICN will more holistically support what is required of a full-fledged Internet mechanism, opening a whole new set of scenarios which are not feasible under the current Internet architecture.

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

1.2 Research Framework

Although ICN's intended target is content distribution scenarios [9], it has been pointed out that the functional basis of this novel networking concept could be applicable beyond such scenarios. In particular, it can be argued that ICN concepts and functionalities, partially match IoT motivations and requirements, thus making this clean slate approach a suitable candidate for an IoT supporting network infrastructure. Leveraging the intrinsic features of ICN could enable compelling solutions to current open challenges, as well as pave the way for new utilisation scenarios. In doing so, ICN concepts will have to be reformulated to account for the stringent requirements commonly associated with IoT scenarios (e.g., energy consumption, emergency notifications, device heterogeneity) [19]

In this context, it becomes paramount to explore how IoT deployments are impacted by the underlying mechanisms of ICN, evidencing not only the expected associated benefits but also identifying its shortcomings, as well as the enhancements to be enforced in order to avoid them. As such, the work in this thesis proposed new mechanisms that evolve ICN concepts surpassing the simple content oriented abstraction of the current ICN paradigm, by designing solutions for the heterogeneous challenges commonly associated with the IoT, encompassing, but not limited to, information management, service and node discovery, interoperability and addressing the full dimension of network resources.

The innovative implementation of ICN concepts, bringing them into the IoT domain, and contributing not only to a more efficient IoT but also to the development of ICN itself, determines the scientific merit of this research which is driven by the pursuit of the following objectives:

1. Provide the vision of an ICN-enabled IoT architecture, materialised through an assessment of the benefits and challenges of utilising ICN as a networking solution for the IoT, determining their relevance in different IoT environments.
2. Extend the core ICN concepts to account for the identified requirements of the IoT scenarios. These modifications should be prototyped and evaluated for demonstrating their practicality and feasibility in such environments.
3. Provide the means for ensuring the interoperability of the proposed solutions to ensure the smooth adoption of these modifications.

1.3 Contributions

ICN is a novel networking paradigm, still at its infancy, initially conceived for content distribution purposes while IoT is an umbrella encompassing a plethora of communication protocols and highly heterogeneous devices and application scenarios. As such, considering ICN as a networking fabric for IoT is not a contained topic, and therefore the contributions of this thesis are not limited to a single topic but are instead spread across different proposals with the common goal of making ICN more suitable to the needs of the IoT. Consequently, one of the significant contributions of this thesis is to provide a comprehensive view of the elements that should be considered for a Named Internet of Things.

The contributions encompassed in this document begin to take form with an overall view on the ICN and IoT landscapes, both as isolated elements and as an integral overview of relevant works aiming at the integration of such technologies. This view is then complemented by introducing the perspectives of ICN for IoT, identifying the main challenges and opportunities associated with the integration of these two technologies [20] and by means of a simulation study making a case for the use of ICN in IoT environments [21]. This study mainly focused on the impact that ICN intrinsic in-network caching mechanisms could bring to the potentially constrained devices taking part in the IoT. As a whole, these results showcased the suitability of pursuing a Named Internet of Things while highlighted the main action points towards its realisation.

As mentioned, one of the key advantages commonly associated to content-centric approaches is intrinsic in-network caching. Associated with this feature, ICN provides a mechanism for controlling how long a given piece of content can be cached before it is considered stale. This mechanism is driven by the information producers, which raises challenges on how to retain the caching related advantages in IoT scenarios involving applications with stringent requirements in terms of "information freshness". As such, an analysis of the implication of these requirements and the associated management mechanisms available in ICN was performed and complemented with a proposal for a consumer-driven information freshness management mechanism [22]. The goal behind such approach is to satisfy the consumers' needs while mitigating the negative impact of stringent information freshness requirements in the overall network performance. This mechanism is further discussed and integrated within a secure management framework, leading to the concept of a Freshness-based Service-Level Agreement (SLA) that allows the retrieval of information as fresh as agreed with the service provider [23].

A different approach for addressing the information freshness issues associated with Interest-based ICNs, mainly in scenarios in which the information generation time is uncertain (e.g., emergency notifications), is to support a publish-subscribe approach on top of Interest-based ICNs. This communication strategy enables not only the retrieval of current information while reducing network overhead and delay as compared to a polling approach, but also enables the push of information which is not natively supported in Interest-based ICNs. The proposed protocol, known as Named Publish

Subscribe Networking (NPSN) [24], is based on its IP counterpart Message Queuing Telemetry Transport (MQTT) and proved to be a feasible and suitable solution in scenarios in which content generation time is highly variable.

The NPSN protocol was further extended with discovery functionalities, which relied on Named Data Networking (NDN) caching and naming features to publish versioned discovery information. The proposed discovery functionalities along with the NPSN protocol itself, are leveraged altogether with the interoperability mechanisms proposed in [25], to enable the operation of NPSN in interoperable environments (e.g., with MQTT) [26]. The idea is to contemplate the existence of a mainly IP-based network as well as different ICN instantiations and to ensure a smooth ICN rollout as it transit to later development stages. These concepts are further examined in a mobility environment in [27], where a new network entity is proposed to allow end-users to retain content reachability while moving across network architectures.

The development of appropriate service discovery mechanisms enriched with semantic capabilities for understanding and processing context information is a crucial feature for turning raw data into useful knowledge and ensuring the interoperability among different devices and applications. Taking this into consideration, broker based discovery mechanisms were enhanced with the use of a semantic matching mechanism, based on semantic similarity, for achieving a flexible discovery process [28].

Additionally, a mechanism for enabling node and service discovery within local connectivity IoT scenarios (e.g., multi-sensory M2M environments) was conceived [29]. Concretely, a local area discovery solution is designed for Interest-based ICN protocols without depending on a dedicated infrastructure by relying on Layer 2 broadcast protocols and enabling support for 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. The goal of the proposed discovery mechanism is to enable the discovery of nodes in the local domain (e.g., NPSN brokers) allowing its further integration with other mechanisms for achieving a combined local/global solution as required for global scale IoT deployments.

Finally, the ICN networking layer was enhanced to not only reach for content using names but to specify and retrieve particular elements from within the content itself. This mechanism is proposed as the basis for IoT content (dis)aggregation and is showcased by integrating ICN and Light Field (LF) Imaging for selectively retrieving specific images based on features contained in the images featuring a security surveillance scenario [30]. These concepts were developed under the scope of the SeLF-ICN project[2].

[2] `https://www.it.pt/Projects/Index/4376`

Table 1.1: List of Publications

Type	Year	Name	Venue	Ref.
Book Chapter	2019	Information Centric Exchange Mechanisms for IoT Interoperable Deployment	CRC Press	[24]
Journal	2019	Exploring interoperability assessment for Future Internet Architectures roll out	Elsevier JNCA	[25]
Journal	2017	Internet of Things discovery in interoperable Information Centric and IP networks	Wiley ITL	[26]
Journal	2017	ICN as Network Infrastructure for Multi-Sensory Devices: Local Domain Service Discovery for ICN-based IoT Environments	Springer WPC	[29]
Journal	2016	On the application of Contextual IoT Service Discovery in Information Centric Networks	Elseiver COMCOM	[28]
Journal	2016	A Secure IoT Management Architecture based on Information-Centric Networking	Elsevier JNCA	[23]
Journal	2016	Information-Centric Networking for the Internet of Things: Challenges and Opportunities	IEEE Network	[20]
Conference	2019	Content Retrieval While Moving Across IP and NDN Network Architectures	IEEE ISCC	[27]
Conference	2018	Selectively Accessing Light Field Face Images over Information Centric Networking	IFIP NTMS	[30]
Conference	2014	A case for ICN usage in IoT environments	GLOBECOM	[21]
Conference	2014	Consumer Driven Information Freshness Approach for Content Centric Networking	IEEE INFOCOM Workshop	[22]

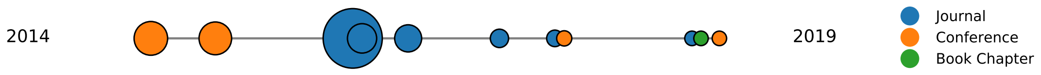


Figure 1.1: Publication timeline

This thesis builds upon different dissemination contributions (i.e., 1 Book Chapter, 6 Journal Papers and 4 Conferences) which are summarised, arranged by type and date, in Table 1.1 and further put into perspective in Figure 1.1, which presents the publication timeline. In the figure, circles represent a publication, and its size is related to the number of citations as by July 2019, as a measure of the impact of the publication in the research community. From the figure, there is a publication that stands out, [20], and as a matter of fact, this work was recognised by the Web of Science[3] as a highly cited paper, being in the top 1% of its year of publication in the academic field of Computer Science.

Besides the listed dissemination contributions, this work has lead to open source software contributions to the reference implementations and tools of the NDN and Content-Centric Networking (CCN) architectures. Also, the libraries and tools developed

[3] https://www.webofknowledge.com/

Table 1.2: List of Software Contributions

Name	Description	References
npsn-cxx	C++ prototype of the Named Publish Subscribe Networking (NPSN) protocol and instances	[24], [26]
pyndnapi	Python library providing abstractions of Consumers and Producers entities and their interaction for NDN.	(Not applicable)
pynpsn	Python prototype of the NPSN protocol and instances	[26]
Plugins for FiFu Framework	Plugins for the FiFu framework for providing interoperability with the protocols: NPSN, NDN and MQTT	[24], [25], [26], [27]
pyselficn	Python Implementation of the SeLF-ICN entities and protocols	[30]
ndn-hosting	Tool for hosting files/folders using NDN	[24], [25]

as part of this PhD work are summarised in Table 1.2.

1.4 Structure

The contents of this document are centred around the two main technologies described in this introductory chapter, namely, ICN and IoT. As such, the corpus of the thesis is composed by a total of six chapters, starting by this introductory chapter. Followed by a background chapter, aiming at positioning the reader in the context of the technical content and innovations identified, proposed and evaluated in the following three central chapters and concluded in a final chapter.

The first two chapters are intended to provide a better understanding of the topics discussed in the thesis document. Chapter 1 motivates and briefly discusses the problem addressed in this work and summarises the main contributions achieved in the development of the concepts and prototypes proposed in the document. Chapter 2 presents an overview of the technologies explored throughout the proposed work, namely IoT and ICN. The vision, concepts and main research challenges commonly associated with the IoT are summarised in that chapter, which then advances to the general concepts of ICN and details the most prominent solutions implementing this networking paradigm. The chapter finalises by discussing relevant works on the integration of both technologies.

Chapter 3 starts by providing key preliminary feasibility experiments showcasing the expected benefits in terms of energy and bandwidth consumption, as well as the flexibility for adapting current ICN caching mechanisms to target specific requirements of the IoT. This experimental technology positioning is complemented with an overview of the perspectives of the application of ICN as a networking infrastructure for the IoT. It outlines the opportunities that ICN brings for addressing the challenges and utilisation patterns commonly associated with the IoT scenarios and environments, as well as on those challenges the IoT imposes to this novel networking paradigm.

The chapter finalises by summarising the central aspects of ICN that require further enhancement to account for the expected IoT scenarios.

The identified challenges are then targeted in Chapters 4 and 5, where solutions are formally proposed. Chapter 4 targets information freshness management from two different perspectives, one based on controlling freshness in polling approaches and another relying on publish-subscribe approaches to receive information as soon as it is generated. Chapter 5 is mainly focused on service and node discovery. Functional requirements, such as security and interoperability, are taken into consideration. Both chapters provide the implementation details of the proposed solutions, along with an assessment of the feasibility of its realization and a detailed discussion on the evaluation results.

The findings and achievements of the entire thesis are summarised in Chapter 6. The chapter puts the contributions of the thesis work into perspective and establishes the suitability of ICN for enabling the IoT. In doing so, the chapter discusses how the individual pieces presented along the document could fit together and identifies open issues and ideas towards a fully-fledged **"Named Internet of Things"**.

CHAPTER 2

Things, names, and naming things

> *“We are all now connected by the Internet, like neurons in a giant brain.”*
>
> — Stephen Hawking

The present chapter introduces the basic concepts to be used throughout this thesis work. It begins with an overview of IoT, as an umbrella covering the technologies related to the expansion of the Internet into the physical realm through the widespread deployment of spatially distributed devices with embedded identification, sensing and actuation capabilities. This vision is complemented with an identification of the primary networking challenges and protocols commonly associated with the IoT landscape.

The fundamental concepts of the ICN paradigm are also discussed, and the main design features of the current solutions are presented. Finally, current efforts concerning the integration of the ICN and IoT concepts are summarised, hinting that the IoT concept could be adequately supported from an ICN perspective.

2.1 Internet of Things (IoT)

The IoT is an emerging technology which has been in the focus of the research community for several years now. Although the term was first introduced by Kevin Ashton in 1999 [31] and the definition of "things" has evolved with the technology development, the main goal of sensing information and acting in consonance, without human intervention, remains the same. The basic idea behind the concept lays on the capabilities of devices to communicate and to interact with the surrounding environments, either by sensing (i.e., sensors) information or triggering a particular tangible action (i.e., actuators). By leveraging the specific characteristics of such devices, it is possible to reduce the gap between the digital and physical worlds. This has prompted the appearance of various applications where the controlled interaction of such devices in more complete platforms turn the simple functionalities of single elements into a complex smart scenario (e.g., smart metering, smart healthcare, smart transportation). Notwithstanding, reaching the full dimension of the IoT, where the scope of these smart scenarios reaches worldwide dimensions, remains a challenge.

2.1.1 Vision and concepts

The proliferation of increasingly small devices with sensing and actuation capabilities, and embedded communication and computation resources, has led to the vision of highly dynamic, distributed and heterogeneous network systems, where these "things" are interconnected, forming the so-called IoT [12]. The possibility to remotely access sensing data from a variety of sources, and to act accordingly opens a whole new set of use cases and research opportunities [14]. For example, by reading environmental parameters captured by sensors such as temperature, relative humidity and air pressure, luminosity and leaf wetness, it is possible to automatically start or stop irrigation and consequently to improve crops production. This example expands into many different areas of impact such as Smart Agriculture, Smart Factory, Smart Cities, Smart Parking, Smart Traffic, Smart Environment, eHealth, amongst others [32]. However, realising such a vision will rise diverse challenges to the supporting networking architecture and substantial advances will be required in various ICT fields [13].

The device miniaturisation, the spatial spread expected to be reached, the reduced costs to ensure massive adoption, are just some design criteria shaping the development of IoT devices. As a result of the different design choices, IoT networks are expected to involve a vast number of heterogeneous devices, in terms of communication and computational resources, type and structure of the produced information, among many other differentiating factors.

The presence of resource-constrained devices is, however, one of the elements shaping the IoT technologies. Having battery-powered devices, with low computational power, expected to be operational for long periods, imposes the utilisation of energy efficient protocols and technologies, as well as long sleep cycles that are considered to extend the battery duration [17, 16]. These are elements that fundamentally differentiate IoT technologies from the solutions for traditional networks forcing strict requirements (e.g., the use of low Maximum Transmission Units (MTUs) and packet sizes) and challenging communication protocols (e.g., multi-hop networks should account for the duty cycles of the devices). Moreover, the complexity of maintaining large amounts of devices, the need to keep them operational in highly mobile environments and the security requirements of sensitive sensor information and actuators operations motivate the need for minimal human intervention and have led to a complete set of self-* functional requirements [18]. Based on the above considerations, the next section summarises the primary networking challenges that need to be addressed in order to turn the IoT from a concept into a well-engineered, commercially viable technological paradigm.

2.1.2 IoT networking challenges

This section summarises the main networking challenges commonly associated with the IoT.

- Heterogeneity: IoT protocols and solutions must account for devices with very distinct capabilities from the computational and communication standpoints.
- Scalability: To support a vision where billions of connected devices exchange information globally, raises scalability challenges including aspects such as naming and addressing, data communication and networking, information and knowledge management, and service provisioning and management.
- Energy Efficiency: Resource-constrained devices have severe limitations in terms of power, communication and computing capabilities. Notably, energy is a scarce resource and, in many IoT scenarios, devices are battery powered and require to be operational for prolonged periods. Therefore, energy-efficient operation design is crucial for any IoT networking solution, and the development of solutions optimising energy usage, even at the expenses of performance, will become increasingly attractive.
- Security and Privacy: By crossing the gap between the physical and digital realms, IoT solutions are tightly entangled with the physical world. Preserving the privacy and providing relevant security mechanisms, to protect the potentially sensitive data exchanged in IoT scenarios, are a crucial requirement for ensuring acceptance by users.
- Quality of Service: QoS requirements can be very different according to the specific scenario being addressed. Some sensing data may require to be timely

received (e.g., emergency notifications), while others may tolerate longer delivery delays (e.g., comfort parameters sensing in a smart home environment). Some IoT applications should also account for data freshness needs, for example when consumers are interested in the latest instance of a frequently updated content (e.g., remote vital signals sensing in eHealth applications), versus an available older copy in a nearby cache point (e.g., room ambient temperature).

- Mobility: IoT scenarios could be, in general, highly dynamic, involving highly mobile devices, which have to smoothly retain connectivity (e.g., car sensor, wristbands). As such, mobility support is a key requirement for IoT protocols.
- Self-* capabilities: The complexity and dynamics of IoT scenarios call for the need for least human intervention in the deployment and maintenance of IoT devices. Moreover, many small embedded devices provide minimal interfaces for management and configuration. Consequently, smart objects can autonomously react to a wide range of different situations, including but not limited to the ability to perform device and service discovery, to build overlays, to adaptively tune protocols behaviour and to adapt to the different contexts.
- Semantic interoperability: The device heterogeneity also affects the way information is structured and disseminated. Despite the efforts to define and utilise standardised formats and models, informational silos still remain a problem. Alternative fuzzy mechanisms based on semantic similarity have been proposed. Breaking the information silos by providing semantic interoperability is a key aspect in order to achieve the truly global dimension of the IoT.

2.1.3 Overview on the IoT stack of protocols

The previously described challenges have been tackled in several research works [33], leading to the emergence of numerous protocols, at different layers, to account for the particularities of the IoT [34]. This section provides an overview of the current shape of the IoT stack of protocols ranging from the link to the application layer.

Low-Power Wireless Personal Area Networks (LoWPAN) relates to low-cost communication networks, enabling wireless connectivity in applications with limited power and relaxed throughput requirements. A reference wireless solution to realise these networks by providing low-power, low-complexity, low-data-rate, short-range wireless data communications in Wireless Personal Area Network (WPAN) is the IEEE 802.15.4 protocol [35]. This protocol enables constrained devices to intercommunicate, with considerably reduced battery consumption, and to send low-bandwidth data to a centralised device. Although the standard merely defines the physical and Medium Access Control (MAC) layers, it has been leveraged by different protocols stacks (e.g., Zigbee

IP, Zigbee Pro, Zigbee RF4CE[1], 6LoWPAN [36]). Additionally, Low Power Wide Area Networks (LPWAN) are an attractive solution to provide large coverage wireless technologies while sharing the low bandwidth, possibly very small packet and application layer data sizes and long battery life operation motivations of the LoWPAN [37]. The leading LPWAN technologies being considered in the IETF are Sigfox[2], LoRaWan[3], Narrowband IoT (NB-IoT) [38].

At the network layer, 6LoWPAN [36] provides header compression and link layer fragmentation to provide IPv6 support on top of constrained link technologies [39]. Additional protocols have been proposed to provide additional functionalities in Low-Power and Lossy Networks (LLNs) such as IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [40], neighbor discovery optimisations for 6LoWPANs [41], Multicast DNS (mDNS) for resource discovery [42] and Datagram Transport Layer Security (DTLS) for providing communication privacy over datagram protocols [43].

From the application-level perspective, MQTT [44], Advanced Message Queuing Protocol (AMQP) [45]; and Constrained Application Protocol (CoAP) [46] are the most relevant solutions for M2M applications [47]. MQTT is a publish/subscribe lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. AMQP is a lightweight messaging protocol designed for a reliable exchange of messages between two entities. CoAP is a specialised web transfer protocol for use with constrained nodes and LLNs that follows a request/response interaction model, thus enabling REpresentational State Transfer (REST) [48] alike communications in constrained environments. CoAP's specification includes support for functionalities such as proxy, caching and discovery.

2.1.4 Limitations of current IoT approaches

As established in Section 2.1.2, IoT raises concerns at multiple levels. The explosion of devices with connectivity capabilities generating and exchanging information challenges the scalability of the supporting networking architecture, and applying a common network functionality may question the "thin waist" approach at the basis of IP networking [49]. The capacity of alternative IP solutions to cope with the huge amount of connected devices is taken to the limit from its addressing to the service provisioning and management. Even content retrieval mechanisms such as Peer to Peer (P2P) and Content Delivery Network (CDN) are subjected to complex issues. For example, caches and proxies break the end-to-end connections assumed by the current security protocols hindering information protection.

[1] `https://www.zigbee.org`

[2] `https://www.sigfox.com`

[3] `https://www.lora-alliance.org/`

Moreover, security is critical in IoT environments due to their close interaction with the physical world. In IP, security was not conceived by design. Instead, it follows the patching approach used to provide increased functionalities to the base IP protocol. In the seek of content security, privacy and data integrity, certain aspects were overlooked (e.g., LLN and constrained devices) and can hinder the performance of existing protocols. IP-based security protocols (e.g., TLS/SSL, DTLS), even though discussed as solutions in 6LoWPAN impose high overhead on the IoT devices in terms of communication operations and resource consumption. In reality, it is the communication channel, between a specific pair of communication nodes, that is secured, rather than the content. Channel-based security is not recommendable for IoT solutions, as it leads to increased overhead in establishing the secure channel and to more resource demanding communication as channel states have to be maintained.

Current energy-efficiency approaches are not handled at the network layer, instead they work at the MAC layer or above the transport layer. For example, CoAP provides a web framework, realised through a subset of REST primitives, however it requires that the devices support a full web stack implementation, which might be prohibitive for some constrained devices. Moreover, strategies such as header compression done in 6LoWPAN can imprint expensive processing requirements over low powered devices.

Mobility support is another key requirement for IoT devices. Although IP mobility management solutions (e.g., Mobile IP [50]) have been continuously under research, they have been commonly associated with scalability problems, leading to more efficient solutions (e.g., Distributed Mobility Management [51]), which have yet to reach adoption by mobile operators. In any case, the validity of such approaches in IoT scenarios is still questionable.

IP networks apply QoS through the execution of different extensions done over the base protocol. However, in general, these extensions work under the principle that resources are reserved at each hop between the source and the content requester, thus requiring extensive signalling, flow identification, and queue processing at the forwarding entities. This is manifested in terms of complexity in routers, which could be further affected by its exposure to the IoT traffic coming from an unprecedented amount of connected devices with the most heterogeneous characteristics and traffic requirements.

2.1.5 Critical Analysis

After examining the most common IoT stack of protocols, it can be established that it is mainly based on the use of TCP/IP protocols. However, the TCP/IP protocols had to be adapted to fit the IoT scenarios. In doing so, many of the functionalities (e.g., caching, security, naming) had to be handled at the application layer. The final goal is to provide a reliable data exchange mechanism for IoT applications. The similarities that can be established between the effective IoT stack and the ICN principles and motivations showcased the potential that this novel networking paradigm could bring to the IoT by addressing its challenges and requirements right at the networking layer. As such, the next section provides further clarifications on the key principles associated with ICN.

2.2 Information-Centric Networking (ICN)

As referred in Chapter 1, the Internet has been continuously evolving. At this point in time, the utilisation scenarios are far from the original intention of interconnecting a reduced set of computers. This change in the way users see the network comes along high demands in terms of reliable, scalable, secure and efficient content distribution. To cope with the new requirements placed upon the Internet two conceptually different approaches have been proposed. First, evolutionary approaches, which propose to patch the Internet in order to provide solutions to the emerging requirements. Second, clean slate approaches, which propose to radically change the foundations of the Internet to accommodate current and envisioned challenges at the design stage, in an Internet built from scratch. This section introduces the main concepts behind the ICN paradigm, a clean slate approach that emerges as an answer to the evolution of the Internet.

ICN proposes to have named content at the centre of the networking fabric, moving away from the host-centric paradigm found in the current Internet protocols towards a content-centric approach in which users specify what information they want to get instead of where to get the information from [9, 10]. The ICN approach was pioneered in TRIAD [52] and has been explored by several research projects (e.g., Data-Oriented Network Architecture (DONA) [53], CCN [54], Publish-Subscribe Internet Technology (PURSUIT) [55], Network of Information (NetInf) [56], NDN [57]). Remarkably, ICN concepts have not remained still but instead have been continuously developed along the years as a result of significant efforts from multiple academic, industrial and standardisation research activities. The ICN architectures, in general, leverage in-network storage for caching, multi-party communication through replication, and interaction models that decouple senders and receivers. The common goal is to achieve efficient and reliable distribution of content by providing, right at the network layer, a general platform for communication services that are today only available in dedicated systems such as P2P overlays and proprietary CDNs.

2.2.1 Interest-based ICNs

Despite existing ICN solutions share some core concepts of this novel paradigm, different implementations follow different design choices (e.g., communication model, naming principles, routing and forwarding). This work will be focused on Interest-based ICN solutions (e.g., NDN, CCN, Community ICN (CICN)). The reasons behind this selection include the active development, variety of implementations, and continuous evolution of these solutions. Moreover, Interest-based ICN already reassembles, at the network layer, the behaviour found in IoT application protocols such as CoAP [18], which makes them a suitable candidate for drafting the solutions presented in this thesis. Notwithstanding, some of the contributions hereafter presented could be adapted to other ICN solutions or even retrofitted in current Internet protocols.

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 2.1 illustrates the communication process in these architectures, along with the main elements composing an Interest-based ICN node. The procedure is handled as follows. Consumers initiate the communication by issuing an Interest. Interests are forwarded towards an entity holding the content, according to the information stored in the Forwarding Information Base (FIB) and following 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 requests 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 are signed by the producers, ensuring both integrity and authenticity of the content and can be cached in the Content Store (CS).

The two more prominent examples of Interest-based ICN protocols are NDN [57] and CCN [54]. The NDN project[4] originally used CCNx[5] as its codebase. In August 2013, the NDN team released NDNx version 0.1, a fork of CCNx version 0.7.2 where "ccn" was renamed to "ndn". From that point forward both projects have followed a different roadmap[6]. While NDNx remained open source, CCNx 1.0 was only released under the PARC Software License in 2016. Later, in 2017, Cisco announced the acquisition of the CCN platform from PARC and the creation of an open-source project within the FD.IO community in the Linux Foundation, known as CICN. Cisco claimed that the acquisition intention is to foster the convergence of different ICN solutions, CCN and NDN in particular, into a harmonised ICN. The ultimate goal is to promote wider and faster adoption of ICN-based solutions as required to overcome the gaps in the

[4] https://named-data.net/project/

[5] https://blogs.parc.com/ccnx/

[6] http://named-data.net/codebase/platform/moving-to-ndnx/

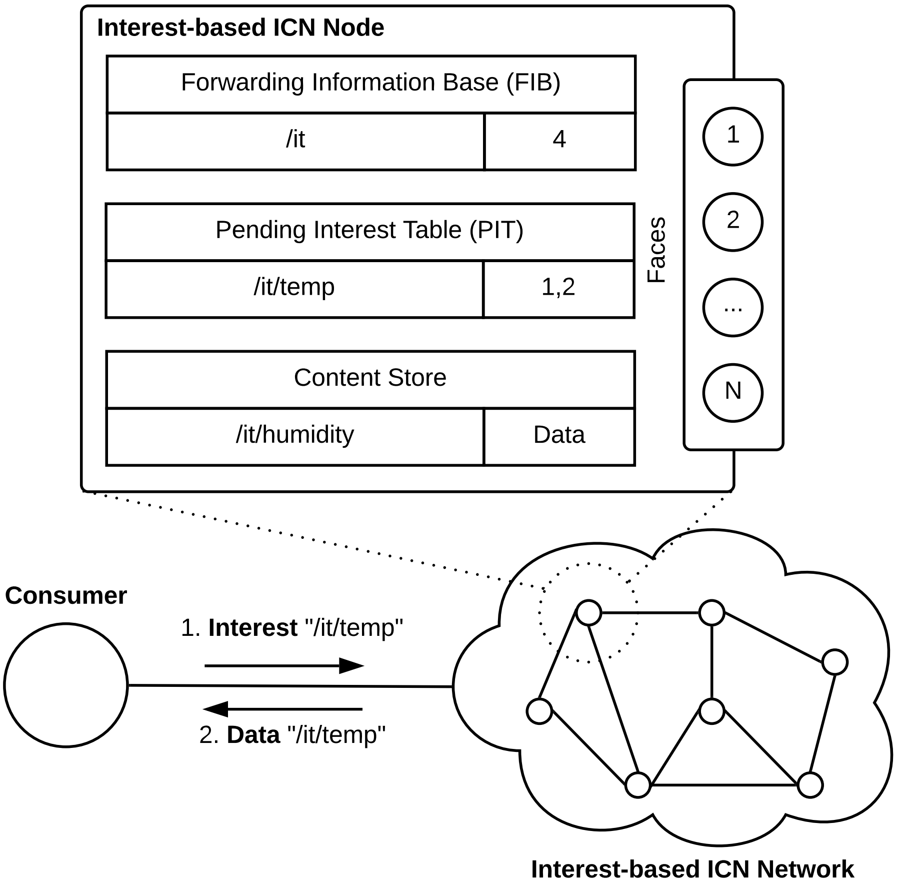

Interest-based ICN Node
Forwarding Information Base (FIB)
/it
4
Pending Interest Table (PIT)
/it/temp
1,2
Content Store
/it/humidity
Data
Faces
1
2
...
N
Consumer
1. **Interest** "/it/temp"
2. **Data** "/it/temp"
Interest-based ICN Network

Figure 2.1: Forwarding in Interest-based ICN networks

current Internet and to satisfy the current and future Internet demands. The differences between these two reference solution are currently the object of a convergence effort at the IRTF [58]. Nevertheless, the maturity, the continuous maintenance and development of the protocol and the existing open source platform, the availability of simulation tools among many other reasons motivated the selection of the NDN architecture for the design and prototyping of the solutions presented in this document. The following section will provide an overview of the foundations of this ICN architecture.

2.2.1.1 NDN Architecture Overview

Named Data Networking is a research project from the *National Science Foundation* which focuses on a Future Internet Architecture. Based on the original code of CCNx, NDN produced a derivative architecture and open-source implementation, sharing the key architectural principles of ICN.

NDN enables receiver-driven communication based on Interest and Data packets. Interest packets are used by consumers to request for content while Data packets are used by any entity in the network to reply back with the requested content when it is available in its cache or if it is a producer of the information.

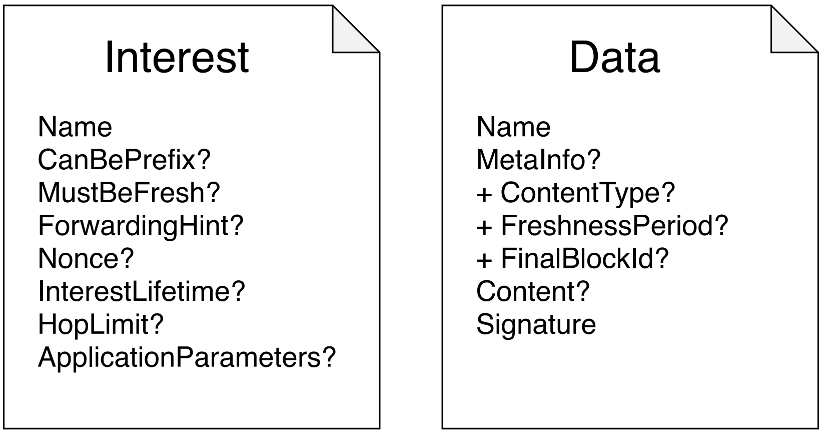


Figure 2.2: Packets format in the NDN architecture

NDN packets are encoded in Type-Length-Value (TLV) format to provide the flexibility of updating the different types as the protocol evolves, as well as to reduce the header overhead in very small packets. As such, NDN packets are actually a collection of nested TLVs and Interest and Data packets are identified by the first of the TLVs in the packet. There is also no packet fragmentation support at the network level, instead, NDN packets may be fragmented and reassembled hop-by-hop. The structure of Interest and Data is represented in Figure 2.2.

Both the packets contain the name of the content being addressed. In NDN, names are hierarchical, encoded using a 2-level nested TLV, and is composed by a sequence of name components. The remaining fields in Interest and Data Packets may be summarised as follows:

Interest Packet:

- CanBePrefix: indicates whether or not the Name contained in the packet may represent a prefix of the desired content or if instead, it provides the exact name of the Data Packet.
- MustBeFresh: indicates whether or not the Interest could be satisfied with staled Data (see FreshnessPeriod in Data packet description).
- ForwardingHint: contains a list of name delegations to be used as part of the forwarding process.
- Nonce: a randomly-generated 4-octet long byte-string which in combination with the name should uniquely identify an Interest packet to avoid Interest loops.
- InterestLifetime: indicate the time that an Interest is held at a given node before it times out; this is a relative time and may be updated by intermediary nodes to account for the forwarding delay.

- HopLimit: indicates the maximum number of hops an Interest is allowed to be forwarded.
- Application Parameters: carries additional information relevant to the Data retrieval process; it must be used in conjunction with an Interest parameters digest component in the content name to ensure the uniqueness and integrity of the parameterised Interest.

Data Packet:

- MetaInfo: includes the following parameters:
  - ContentType: specifies the type of content being carried (e.g., BLOB, LINK, KEY, NACK).
  - FreshnessPeriod: indicates how long a given data may be held at a node before marking it as "staled" or "non-fresh", this field is used in conjunction with the MustBeFresh field in Interest packets with the goal of obtaining new information that may have been created at the source.
  - FinalBlockId: identifies the final block in a sequence of fragments, and should be present in the final block itself, but may also be present in other fragments to provide, in advance, this information to consumers.
- Content: the actual payload of the packet.
- Signature: includes consecutive TLVs: (i) SignatureInfo, which provides the relevant signature information (e.g., signature algorithm, key locator) and is included in the calculation of the signature; and (ii) SignatureValue, which is excluded from the signature calculation and provides the actual bits of the signature.

The exchange of these packets is driven by two main aspects. The first one refers to the included name of the content, which allows it to be uniquely identified. The second is the set of data structures existing in NDN-enabled routers, namely the PIT, which stores requests for content that have not been replied yet; the FIB, which maps names to interfaces; and the CS, which acts as a temporary cache for content that is replied back to requests. This unique combination allows the requested content to be decoupled from where it was obtained, supported by security measures applied directly to the content instead of the communication channel.

The NDN Platform[7] is the reference open source platform of the NDN architecture and considers a node to be composed by different *Faces* (i.e., a generalisation of an

[7] https://named-data.net/codebase/platform/

interface that may represent either a physical interface, an overlay tunnel or a UNIX-domain socket to a local application). The NDN Forwarding Daemon (NFD) [59] is a network forwarder that implements and evolves together with the NDN protocol. The different Faces communicate with each other through the NFD, which maintains internal data structures such as CS, PIT, and FIB, and implements the packet processing logic. The NDN platform also includes the NDN Common Client Libraries (NDN-CCL) which are available in C++, Python, JavaScript and Java. These libraries provide a common API for applications to use NDN and enable an application to send interests to and receive data from an NFD.

2.3 Information Centric Networking in the Internet of Things

In the previous sections, the major concepts ensuring the adequate interpretation of the IoT vision and the ICN approaches in the context of the present work have been identified. Also, the suitability of applying ICN concepts to IoT scenarios has been hinted. The current section aims at grasping the principal efforts of the research community in evaluating the information-centric approaches with respect to their capacity to handle the requirements imposed by the envisioned IoT scenarios.

The recent developments in the field of IoT and ICN have triggered an increasing interest of academic, industry, and standardisation players in exploring ICN as a networking infrastructure for the IoT. Particularly, the ICNRG 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 [11]. A detailed analysis identifying the main benefits, challenges and design choices for efficient and scalable addressing of IoT scenarios from an ICN perspective is provided in [19, 20]. 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.

Named content and its identification right at the networking layer is definitely identified as a key feature making the case for an information-centric approach for the IoT. It enables more efficient coordination and collaboration of actions among peers, as well as the enrolment of the networking elements in tasks such as in-network processing. This fact is reinforced in [60] which proposes a software architecture supporting named data and in-network processing within an operational multi-application sensor network. Using context information such as sensor type and location at the lowest communication levels allows to perform in-network processing with filters, supporting data aggregation, nested queries and similar techniques which are critical to reduce network traffic and save energy.

The high amount of nodes expected to take part in the IoT scenarios not only rises scalability challenges, but also crucial problems regarding the heterogeneity of devices taking part in the IoT [61]. Particularly, the existence of large amounts of network devices with constrained resources in terms of power consumption, storage, computing capability and communications is expected. This issue has been coherently addressed in [62], where Song et al. propose a content-centric internetworking scheme for constrained devices in which, based on the key ideas and basic models of CCN, overcapacity tasks are mapped into stronger devices.

Aiming at grounding NDN architecture research through experimental deployment, Burke [63] distinguishes two emerging application areas that embody the potential of the Future Internet and incorporate the IoT; the Media-rich instrumented environments, which realise the vision of experiential cyber-physical systems that began with ubiquitous computing; and the Participatory sensing, which enables globally-scaled but personally regulated sensing for science, health, planning, and expression. In doing so, advantages derived from the interconnection of sensors and actuators based on NDN are outlined, namely, Internet-scale and wire-speed routing based on names; IP-independent addressing; no distinction between media and control data; and integrated support for security and trust. Namespace query and propagation and service discovery are identified as challenges that still need to be overcome.

Burke et al. [64] also studied lighting control over NDN, which in turn is an example of an "instrumented environment". This application explores a new scope for ICN, which in the context of large scale content dissemination is generally discussed as opposed to control, actuation, or remote execution. Naming reflects fixtures with evolved identification and node reaching capabilities, thus simplifying bootstrapping, discovery, and user interaction with nodes. The authors consider that an ICN-based system requires less maintenance and troubleshooting than typical IP-based alternatives.

Biswas et al. [65] visualise ICN as a contextualised information-centric bus (CIBUS) over which diverse sets of service producers and consumers co-exist. The authors outline several features provided by the content-centric approach that make it suitable for different application environments such as a home network (home-net), Vehicle to Vehicle (V2V), and IoT. They argue that these features make ICN suitable not only for existing applications such as content distribution but also for emerging applications such as sensor or ad-hoc networks. As a result, ICN is leveraged to unify different platforms to serve consumer-producer interaction in both infrastructure and ad-hoc settings. Ravindran et al. [66] particularise how a content-centric approach may be applied in a home network environment, and emphasises the need for a homogeneous platform to handle the diversity of devices, services, and user needs. In this context, name-based protocols are developed to enable zero-configuration node and service discovery, contextual service publishing and subscription, policy-based routing and forwarding with name-based firewall, and ad-hoc device-to-device communication.

In [67], Dinh et al. propose the ICN approach as a fundamental driver for Wireless Sensor and Actuator Networks (WSAN). The authors leverage the capabilities of ICN and introduce some changes for achieving efficient coordination, interoperability, service discovery, and prioritised routing. A novel continuous mode for Interest packets, Continuous Interest (CI), is proposed. CI packets are not intended to be deleted after a corresponding Data packet has satisfied the Interest, and consequently, their lifetime is set to higher values, as compared to the conventional CCN Interest packets. In other words, once a CI packet reaches a producer, it is valid until its lifetime expires or a corresponding explicit cancel message is received. A Data packet is generated and forwarded back to the corresponding requester if there are some events or changes in the sensing data. This exploration establishes a potential foundation to improve performance of WSANs while opening new avenues for future research in ICN-based IoT. Saadallah et al. [68] go forward in the area of ICN-based WSAN by implementing and integrating a CCN communication stack into the Contiki[8] operating system used for resource-constrained embedded systems and wireless sensor networks. The implementation is evaluated under varying network sizes conditions through simulation and real deployment on a testbed.

Ren et al. [69] design, implement and evaluate CCN-WSN, a lightweight variant of a CCN protocol specific for Wireless Sensor Networks (WSN). In doing so, the CCNx protocol was modified for dealing with the memory and computational resources constraints associated to sensor nodes and communication patterns in WSN (e.g., simplified message format, a flexible naming for including data within Interest messages as a name component). Authors claim that using CCN right on top of IEEE 802.15.4 provides an efficient solution and reduces overhead. Since security and privacy were not considered as part of the CCN-WSN design, signature and all security-related fields are omitted, letting this issue for future work. In dealing with the challenge of efficient data aggregation in WSN, Teubler et al. [70] extend CCN-WSN [69] by introducing three major components: unicast faces to reduce the number of message in a broadcast medium; a forwarding service to create overlay structures on a given physical topology; and an intra-node protocol for communication between applications and the forwarding service. The daemon and the interfaces of CCN-WSN remained unchanged in this approach. The application used for evaluating the proposed solution was a simple WSN node discovery which concatenates the received lists of node's ID and adds its own node ID.

[8] www.contiki-os.org

Wang et al. in [71] and [72] explore an ICN approach for direct V2V communications, identifying and addressing issues such as the need for data pushing instead of pulling, and the need of well-defined application naming conventions, understandable for all the vehicles and flexible enough to allow vehicles to express exactly what kind of data they may desire. The capacity to push information is also explored in [73] and in [74], which also targets aspects related to multi-source data retrieval.

ICN's in-network caching mechanism makes content available anywhere on the network, therefore obsoleting traditional Internet security mechanisms based on establishing secure point-to-point channels. In this context, the use of authenticated Interests (i.e., include security information embedded within Interest's name) and encryption-based access control (i.e., encrypt content to prevent its invalid disclosure and/or modification) has been studied for ensuring secure actuation [75] and sensing [76] in IoT-like environments.

Authors in [77] perform an experimental analysis of the impact of ICN applied to IoT. Their work showcases 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. Finally, inspired by 6LoWPAN, a convergence layer for Interest-based ICNs over IEEE 802.15.4 LoWPAN networks is discussed in [78].

2.4 Summary

The integration of information and communication technologies into common devices has been at the genesis of the Internet of Things (IoT). Affecting different sectors and placing disparate new requirements over existing protocols. On the one hand, this has simplified new radio solutions for IoT, such as Zigbee, LoRa, SigFox and others, with electronics miniaturisation providing small-factor interfaces. On the other hand, the Internet networking Protocol has remained crystallised, with few innovations (i.e., 6LoWPAN, which conveyed encapsulation, header compression and resilience mechanisms) not actually providing any new benefit for IoT at a worldwide scale. In fact, IoT has substantially been relying more on protocols above the Transport layer, such as MQTT, CoAP, AMQP, reflecting a more web-based services approach to device reachability and information consumption.

Parallel to this, efforts have been made in exploring disruptive new approaches to network layer design, such as ICN. NDN is a prominent ICN instantiation that decouples the content's identification and its respective location provides data caching at every network routing-able element and intrinsic security. These characteristics have placed this new architecture under the scope of new IoT possibilities for the Internet.

Important research works have been developed towards the integration of the ICN and IoT concepts. These efforts, although isolated attempts aiming at providing a solution to specific IoT deployments from an ICN perspective, have drawn the attention of the research community on the feasibility of approaching the IoT vision by applying the ICN concept. In this sense, the contributions discussed above are considered to pave the way for the conceptual understanding of ICN strengths, weaknesses and opportunities when applied to IoT scenarios. The next chapter will provide aim at the identification of the main perspectives of following this vision.

CHAPTER 3

Crossing the digital-physical gap through names

"We cannot solve our problems with the same thinking we used when we created them."

— Albert Einstein

Previous chapters have led to an understanding of the networking challenges associated with the IoT, the limitation of current solutions, and made the case for an information-centric approach to these challenges and limitations. The heterogeneity in technologies, devices and exchanged information; extremely large numbers of devices involved in a global information infrastructure; the need for energy efficient mechanisms; security and privacy-preserving by design are just some of them. However, by bringing features such as naming, in-network caching, content-based security and interaction models that decouple senders and receivers into the network layer, the ICN paradigm promises to provide many benefits to existing IoT scenarios, as well as enabling novel ones. This chapter motivates the usage of ICN for IoT by means of feasibility experiments which are then complemented with an analysis of the perspectives in terms of challenges and opportunities of the adoption of this concept. The chapter finalises by providing the main guidelines that defined the course of the solutions provided in the remaining of this document.

3.1 Initial assessment of ICN in IoT

In order to showcase the improvements that could be provided by ICN in IoT architectures when compared with IP, this section presents some preliminary feasibility experiments. The goal is to provide an initial assessment and validation of the concept towards the ulterior identification of the main benefits and shortcomings associated. The specific evaluation goals defined for these experiments are focused on the assessment by simulation of the following research issues:

1. To provide a quantitative comparison between IP and ICN based IoT environments in terms of the energy consumed by the information producers and the bandwidth requirements. A different number of sensors and consumers actively involved in the studied scenarios should be considered in order to evaluate the scalability of both supporting networking architectures.
2. The impact of the freshness parameter included in NDN Data packets and of the size of the Content Stores (expressed as the fraction of the total generated content that fits into the cache) on energy consumption, bandwidth and in the percentage of cache hit and source retrievals.

3.1.1 Motivation

While content access and delivery become dominant activities and the Internet grows into a significant contributor to the worldwide energy consumption, the support of content distribution in an energy-efficient and scalable way results increasingly critical. One common and important feature of ICN architectures is the leverage of built-in network caches for improving the transmission efficiency in content dissemination (e.g., low dissemination latency, network transport load reduction). Various aspects that affect ICN caching performance include cache dimensioning, application-independent cache space sharing in a single cache node, cache decision and replacement policies and availability of cached contents [79].

Distributed in-network caching is an attractive property of ICN architectures as it avoids inefficient retransmissions of the same data from the source, simultaneously favouring multipoint to multipoint communication. Besides its bandwidth/storage tradeoff [80], ICN in-network caching appears as a promising solution for improving the overall energy efficiency of the network [81]. Numerous research efforts have focused on energy consumption models for ICN, used for evaluating the impact of caching optimisation mechanisms [82, 83, 84].

3.1.2 Setting up simulations

In choosing a simulation tool, the possibility of simulating wireless networks both based on IP and NDN and the completeness of the NDN stack support were key

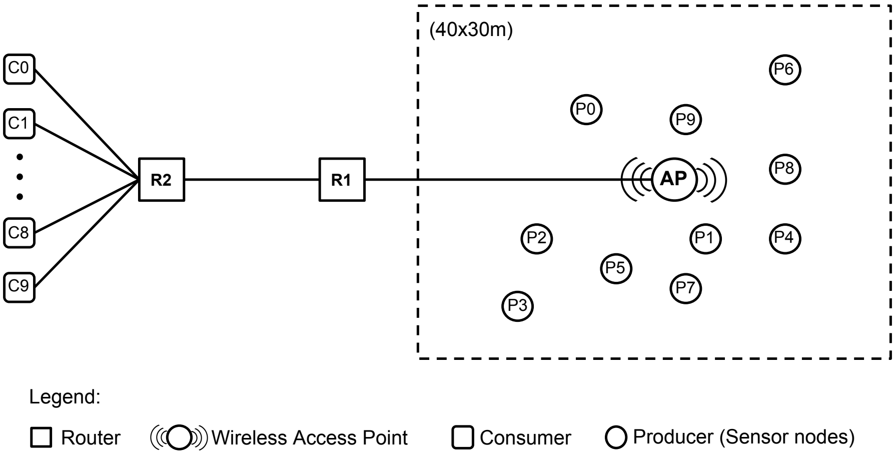


Figure 3.1: Topology of the simulated scenarios

requirements. Finally, the Network Simulator-3 (ns-3), a flexible and easy-to-use tool suitable for wireless network simulation, was selected for conducting this study. For the evaluation of the ICN based approach, the ndnSim [85] simulation tool was selected, and the integrated energy framework for ns-3 [86] was used for modelling the energy source as well as the energy consumption. Additionally, the ndnSIM includes basic traffic generator applications and helper classes that simplify the creation of simulation scenarios as well as tools to ease the gathering of simulation statistics for measurement purposes.

Modifications and extensions to the modules provided by ndnSIM were required to ensure the development of simulations closer to IoT concepts. In particular, a consumer application was implemented in such a way that it always asks for the same content name (e.g., a display screen periodically asking for the latest temperature value from sensors). For the comparison with IP-based scenarios, a UDP based IoT client-server application was created to be consistent with the application used for the ICN-based scenarios. In both cases, the client applications send periodical requests and receive data information in accordance. The information reported by the sensors were defined to have a size of 100 bytes.

The simulated scenarios were designed to facilitate the achievement of the presented research goals and are based on a topology composed initially by ten producers and ten consumers, as shown in Figure 3.1. The number of consumers and producers are varied among simulations. Producer nodes are Wi-Fi (802.11g) stations connected to an Access Point (AP) in infrastructure mode. The AP is connected to a router, which in its turn is connected to the router associated with the consumer nodes. The producers are randomly allocated within a 40x30 meters rectangle centred on the AP. Wi-Fi stations were set to follow a random walk movement within the boundaries of such a rectangle. The Three Log Distance and Nakagami propagation loss models provided by NS-3 were selected to take into account the losses due to variations on distance, as well as the fast fading associated with wireless channels [87].

Consumer applications start and finish requesting content at some random instant chosen uniformly within the intervals $[0; 15]$ and $[45; 60]$ seconds respectively. Each consumer requests sensed data from all the producers at a rate of one request per second with random requesting times uniformly distributed.

For the scenarios that implement the ICN use case, all the consumers were considered to have the same information freshness requirements. Consumers and producers were set up with no caching capabilities to avoid content from being supplied by their internal cache. In all the cases, the LRU (Least Recently Used) cache replacement policy was applied. The Freshness value and Content Store size are varied among simulations. For all the simulations, ten runs were considered and a 95% confidence interval for the obtained results was determined.

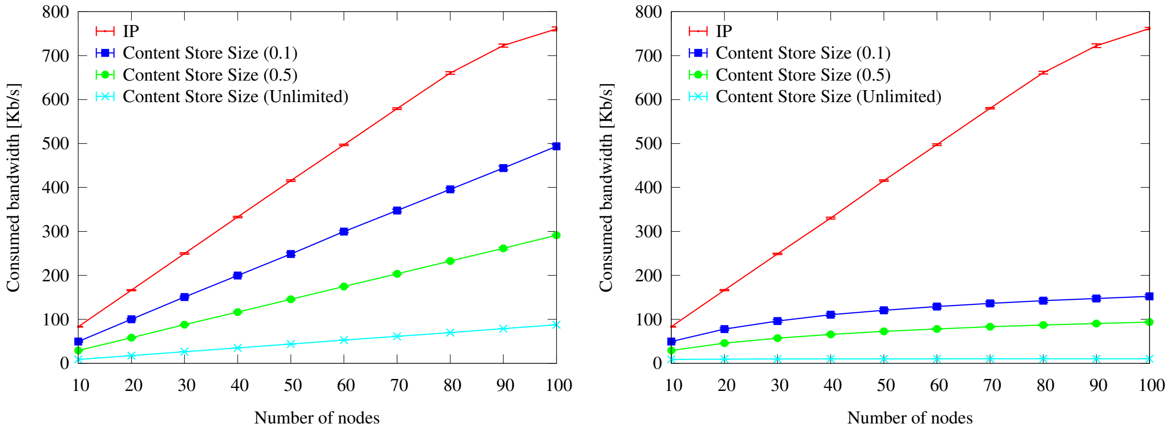
3.1.3 Results and Discussion

The results of the impact in both bandwidth usage and energy consumption in an IP based scenario as well as in ICN-based scenarios with different Content Store sizes are summarised in this section. Although different freshness values were evaluated, the results being presented are those obtained using a freshness value of *t=1s*, as it was the value that imposed more strict requirements to the studied parameters.

Figure 3.2 shows the bandwidth consumed in the link from Router 1 to Router 2 (i.e., downlink direction). In particular, Figure 3.2a involves scenarios where the number of consumers remained constant (i.e., 10 consumers) while the number of sensors was increased from 10 to 100 with a 10 sensors step, and Figure 3.2b considers scenarios where the number of sensors remained constant (i.e., 10 sensors) while the number of consumers was increased in a similar way as in the previous case.

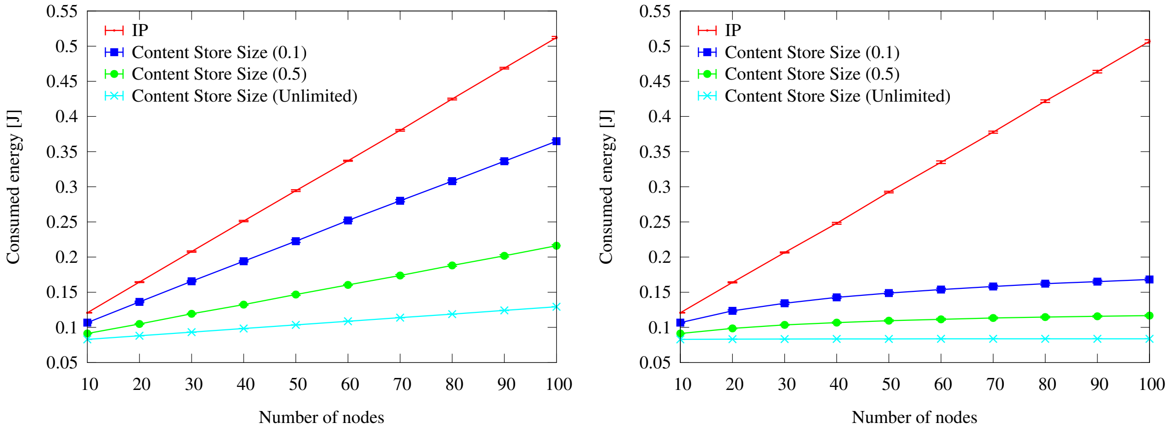
Figure 3.2 shows that for smaller Content Store sizes the potential savings in terms of bandwidth associated with ICN decrease. Nevertheless, the studied ICN-based scenarios outperformed the IP based scenario. The savings associated to the worst case ICN scenario (relative Content Store size of 10%) as compared with IP are approximately 40% when increasing the number sensors (Figure 3.2a) and range from 40.5 to 80% when increasing the number consumers (Figure 3.2b).

Increasing the number of sensors, or increasing the number of consumers, in the IP based scenario, showed to have the same impact on the required bandwidth. A similar analysis for the ICN-based scenarios shows that increasing the number of sensors had more impact on the required bandwidth, as compared with the impact of increasing



(a) Increasing Sensors **(b)** Increasing Consumers

Figure 3.2: Bandwidth consumption in the downlink for different topologies



(a) Increasing Sensors **(b)** Increasing Consumers

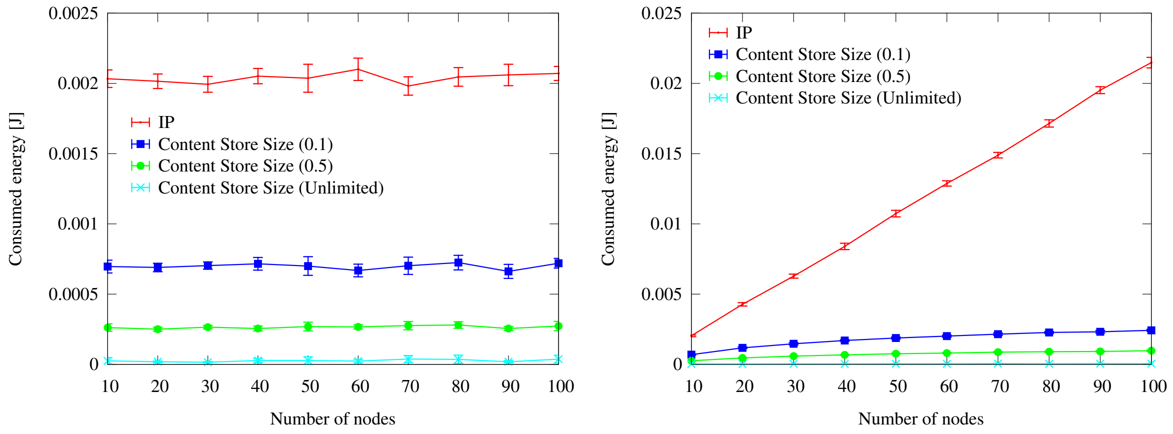
Figure 3.3: Energy consumption at the sensor nodes for different topologies

the number of consumers. This fact is particularly positive considering that in an IoT environment, the number of consumers is not as controllable and predictable as the number of sensors involved in a certain WSAN.

Finally, from Figures 3.2a and 3.2b it can be noticed, for higher numbers of nodes, how the bandwidth consumed in the IP based scenarios loses its linearity because it gets closer to the maximum bandwidth defined for the link.

Figure 3.3 shows the total energy consumption at one of the sensors for different topologies organised as provided for Figure 3.2. Results present a similar behaviour as that identified for bandwidth consumption. The savings of total energy consumption associated with the worst case ICN scenario (relative Content Store size of 10%) as compared with IP, range from 11.6 to 28.8% when increasing the number sensors (Figure 3.3a) and from 11.6 to 66.8% when increasing the number consumers (Figure 3.3b).

Since no power saving mechanism was used, the reflected total consumption not only



(a) Increasing Sensors

(b) Increasing Consumers

Figure 3.4: Fraction of energy consumed at the sensor nodes due to the transmission process

accounts for the communications originated or destined to the sensor being analysed, but also includes the reception of those packets that, given the shared nature of the medium, reach the sensor although it is not the intended receiver.

Going deeper into the energy consumption analysis in IP and ICN based IoT environments, Figure 3.4 shows the fraction of the total energy expended exclusively due to transmissions at one of the sensors for different topologies and organised as provided for Figures 3.2 and 3.3. Since an increment on the number of sensors is not connected with the amount of information to be transmitted by each sensor, the energy consumed due to transmissions is not expected to change. This fact is ratified by the results shown in Figure 3.4a. The savings associated to the worst case ICN scenario (relative Content Store size of 10%) as compared with IP are approximately of 65% when increasing the sensors (Figure 3.4a) and range from 65.8 to 88.7% when increasing the consumers (Figure 3.4b).

Table 3.1 provides an insight into how the freshness parameter and the Content Store size impact the source of the information. The table shows, in percentage, the number of packets directly retrieved from the source and those obtained from the cache, giving particular emphasis to the packets that were obtained directly from the cache of Router 2 (i.e., the first cache in the content retrieval path).

Table 3.1: Percentages of Source and Cache Hit

Content Store Size [%]	**10%**		**50%**		**Unlimited**	
Freshness [s]	**1**	∞	**1**	∞	**1**	∞
From Source	90.24	90.24	41.50	41.00	15.11	0.24
From Cache	9.76	9.76	58.50	59.00	84.89	99.76
From First Cache	9.75	9.75	49.53	49.92	84.89	99.76

Based on the results in Table 3.1, it is possible to establish that:

- More strict information freshness requirements reduce the number of cache hits, thus requiring new data to be fetched directly from the source.
- The effects of the freshness requirements are less relevant for smaller Content Stores.
- Most of the information is provided by the Content Store closer to the Consumers (i.e., the storage capacity of the network is being underutilised).

3.1.4 Lessons learned

These experiments showcased the potential of ICN contributions in addressing IoT scenarios, compared to IP-based mechanisms. Notably, the benefits that the ICN in-network caching mechanism can bring in terms of network traffic reduction and energy saving. The results have demonstrated that by consuming less energy and bandwidth, ICN outperforms IP based scenarios. Results for different Content Store sizes in ICN based scenarios have shown that, as the available caching storage capacity increases, the consumption of both energy and bandwidth get reduced. Based on the identified relevance of the available storage for caching purposes and considering that typically the amount of space required for sensed data is much lower than the one associated to other content types such as multimedia, it is advisable to ensure a dedicated cache for IoT content. In this way, these experiments show that the deployment of ICN in increasingly important areas, such as IoT, can make a positive contribution therein, where IP based designs are otherwise becoming more complex and hard to deploy.

3.2 On the perspectives of ICN for IoT

Motivated by the results of the previous simulation study, the current section presents the key features of ICN, explores the opportunities of leveraging such features in IoT deployments, identifies the potential challenges that the underlying networking operations of ICN may impose over key aspects of IoT, and provides guidelines to address those challenges.

3.2.1 Key features

ICN, as its name suggests, centres its networking functions around content in contrast to the host-centric approach of the current Internet. By identifying content instead of hosts, ICN intrinsically supports content retrieval from any host holding a valid copy of the object, thus decoupling information consumers and producers, which leads to more efficient content distribution, ease in mobility and multihoming and better disruption tolerance [9].

Key ICN features include name-based content retrieval, in-network caching, content-based security, and connectionless communication. Naming is a key enabler of the remaining ICN features, by naming content right at the networking layer, it becomes agnostic of the source of a particular data. Still, it is required to provide proper content security to ensure the authenticity and the integrity of such a piece of data. With these two mechanisms in place, it is possible to store the content anywhere and to retrieve it via basic networking primitives. Moreover, this decouples the producer and the consumer of the information in a connectionless-based communication.

3.2.2 ICN for the IoT: Strengths, shortcomings and guidelines

This section builds upon the relevant related work provided in Section 2.3 to put ICN's key features into perspective and identify the main opportunities, challenges, and research paths towards the adoption of ICN for approaching IoT scenarios.

3.2.2.1 Naming

Information-centric approaches, identify content with names, which are used for forwarding and routing throughout the network. Naming content at the network layer can be further leveraged for performing additional functions such as content filtering (e.g., `/it/room136/sensor03/temperature` is likely to be an IoT traffic involving information from a temperature sensor), in-network processing (e.g., a request like `/it/room136/temperature/average` can trigger multiple requests to get the information from all the temperature sensors in the room and will be responded with the average of all the received values) and content aggregation (e.g., `/it/room136/temperature` could have the aggregated content from all the temperature sensors in the referred room) which are relevant for an efficient IoT. Moreover, the use of names at the networking layer could ease task such as coordination and collaboration among different networking entities easing the implementation of self-organising capabilities for IoT devices.

The IoT is composed of a large number of heterogeneous devices from different manufacturers, which specify their own information sharing structure leading to informational silos [88]. This has hindered the interoperability between different applications and the realisation of more complex IoT scenarios. This problem is further extended to the use of names when considering ICN-enabled IoT scenarios. Therefore, although the use of names may ease the IoT deployment and operation, there is a need to carefully tackle the heterogeneity in terms of naming and nomenclature to refer to IoT content. A possible way of addressing this issue is to provide data with adequate and standardised formats, models and semantic description of their content (metadata), using well-defined languages and formats. 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 [28]. The goal behind 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.

Besides this naming semantics heterogeneity, the adoption of a naming scheme could also be conditioned by the target IoT scenarios. For example, while in some cases it is necessary to uniquely identify a piece of content (e.g., registering historical sensing data, addressing a particular actuator), in other cases it may be desirable to have mutable content (e.g. a live video surveillance stream) [30].

From an operational point of view, by having named content at the centre of the networking functions, ICN is expected to raise other issues when considering the characteristics of IoT traffic. While IoT content is generally small, uniquely identifying it could require relatively long names leading not only to an issue in terms of overhead but also challenging constrained networks with low MTUs (e.g., Zigbee). Also, mechanisms for addressing content yet to be provided (e.g., by describing the content using metadata), have to be implemented as it may be required in specific IoT scenarios (e.g., to force a sensor to perform a sensing operation and reply with the result). This last issue becomes even more critical when considering self-certifying namespaces in which the names depend on the data itself.

3.2.2.2 In-network caching

The possibility of retrieving cached content may be leveraged for avoiding unnecessary transmissions, saving the scarce resources commonly associated with IoT-based devices (e.g., bandwidth, battery lifetime). Also, having the content distributed in the network storage eases content distribution, optimising its delivery by bringing the content closer to the consumers as it is spread over the network.

In-network caching is then recognised to be one of the prime contributions of the information-centric paradigm [79]. However, some IoT scenarios may require close to real-time communications, which not only diminishes the advantages that caching systems may bring to the overall performance, but may also lead to the retrieval of obsolete information. Moreover, in ICN systems involving distributed caching, producers have reduced control over the content replication and therefore, it is technically challenging to find all copies of a given content and mark them as stale. Although some mechanisms have been considered by the different ICN solutions to address this issue (e.g., use of a freshness parameter in NDN Data packets which defines the time a given piece of content can be cached by a given node before it becomes stale), they are still insufficient for a full control of the information freshness (e.g., because of the stackable nature of the freshness parameter, a Data could be retrieved from a given node just before it is marked as stale and be therefore cached again for the same time and then it can happen all over again repeatedly).

ICN caching may be done as the content is transmitted towards the consumer (i.e., on-path caching) or by intentionally placing the content in a specific network element (i.e., off-path caching) [10]. While the former caching solution is simpler and requires presumably no signalling, the second approach requires additional mechanisms to be enforced for managing the caching process, thus requiring additional overhead in terms of communication and processing. Moreover, achieving optimal content retrieval (e.g., identify the closest off-path cached copy of the content) remains a challenge and proactive caching of content (e.g., by relying upon Software-Defined Networking (SDN) [89]) could be considered for achieving even better performance [90]. In terms of security, by having caching at the network layer, ICN must carefully address attacks targeting the caching mechanisms (e.g., pollution and polling attacks) [91].

3.2.2.3 Content-based security

Along with the networking paradigm shift, ICN moves away from the channel security strategy into content-based security [92, 93]. As such, content may be retrieved from untrusted sources and through insecure channels. This fact gains relevance in IoT scenarios because the content could be secured just once and then redistributed throughout the network, saving processing resources with respect to a channel-based security approach in which each consumer should establish an end-to-end secure channel, thus hindering the scalability of a secure IoT [20].

Despite the fact that different ICN solutions intrinsically implement security mechanisms, achieving security levels in compliance with the requirements of the IoT calls for further development. Moreover, the cryptographic algorithms considered by ICN security mechanisms could be too demanding for constrained devices, thus requiring different algorithms (e.g., elliptic curve cryptography) or even securing strategies (e.g., the delegation of cryptographic functions) [94]. Additionally, some security dimensions are more complex to address and have to be redefined from the way they are currently achieved [75, 76, 95]. For example, in ICN it should be assumed that any consumer can access any piece of content (e.g., by retrieving it from a cache), consequently controlling the access to the information should be done differently (e.g., by obfuscating the content itself) [92].

3.2.2.4 Connectionless communications

ICN's features are empowered when considered altogether in a connectionless communication where the consumer and producer of the information are decoupled, which can be largely benefited in lossy wireless environments (statistically fragmented) involving low-powered devices (possibly resource-constrained and with long sleep cycles) commonly found in IoT scenarios [96].

ICN architectures, in general, follow a pull-based communication approach, either Request/Response or Publish/Subscribe. However, specific scenarios may require push-like communications (e.g., notification of a threshold). As such, mechanisms for emulating push-based communications should be defined. Additionally, although ICN networking operation is centred around content, in some cases it may also be desired to address a particular host (e.g., sending an action to a particular actuator, management related communications). A solution to these issues could include approaches in which producers send an Interest expressing their willingness to send Data, thus initiating a communication [73] or otherwise setup a long-lasting communication channel through the use of long-term interests [67] in which consumers keep listening for new Data packets.

3.3 Summary

Despite current efforts to approach the IoT by means of the traditional host-to-host communication paradigm, the "thin waist" of IP based approaches rises challenges at the different networking layers [18]. On the other hand, ICN approaches provide a more natural and flexible way of addressing the challenges imposed by the IoT [20, 19, 97, 98]. Notwithstanding, approaching IoT scenarios from an ICN perspective faces the most dynamic and diverse types of challenges.

In this chapter, a simulation study showcased the benefits of using ICN, highlighting the feasibility of such solutions and the advisability of pursuing this line of research. In this line of thought, the main perspectives for the adoption of ICN as a networking infrastructure for the IoT are summarised. The main strengths and shortcomings of ICN approaches are identified, and guidelines for evolving this novel paradigm towards the provisioning of IoT support are outlined and summarised in Table 3.2.

The vision on the perspectives of ICN for the IoT, summarised in Figure 3.5, considers heterogeneous devices, with different computational and communication capabilities, and connected through different access technologies. Devices could either interact through a gateway or otherwise directly connected to the network. In these environments, the intrinsic functionalities of ICN such as in-network caching and content-based security are leveraged. Also, in this vision, the roll-out of the ICN architectures is considered to be a gradual process and, as such, the coexistence with the current IP based networks is expected [99].

Based on the outcomes of the current chapter, it can be established that there are several aspects which require further enhancements in order to support the envisioned

IoT scenarios and applications from an Information-Centric Networking perspective.

Table 3.2: ICN for IoT: strengths, shortcomings and development guidelines

Feature	Strengths	Shortcomings	Guidelines
Naming	- In-network processing and auto-configuration. - Information filtering and aggregation. - Semantic coordination and collaboration.	- Naming conventions are required. - Immutable vs. mutable object. - Data size vs. name size.	- Support for unstructured information - Hybrid Hierarchical and flat namespaces
Caching	- Optimised content delivery. - Resource efficiency. - Disruption tolerance.	- Consistency of cached object. - Caching decision policies for optimal space usage. - Proactive data pushing.	- Freshness control - SDN solutions for proactive caching
Security	- Secure once approach. - Easy data sharing without involving the original source.	- Resource demanding algorithms - Managing consumer identity and access policies. - Future content in self-certifying namespaces.	- Lightweight cryptographic mechanisms. - Delegation of cryptographic functions.
Pull-based	- Connectionless receiver-driven communication - Decouples information senders and receivers - Mobility and multihoming.	- Support for push-like communications. - Emergency notification. - Management and control.	- Persistent Interests - Interest-Data-Interest-Data sequence

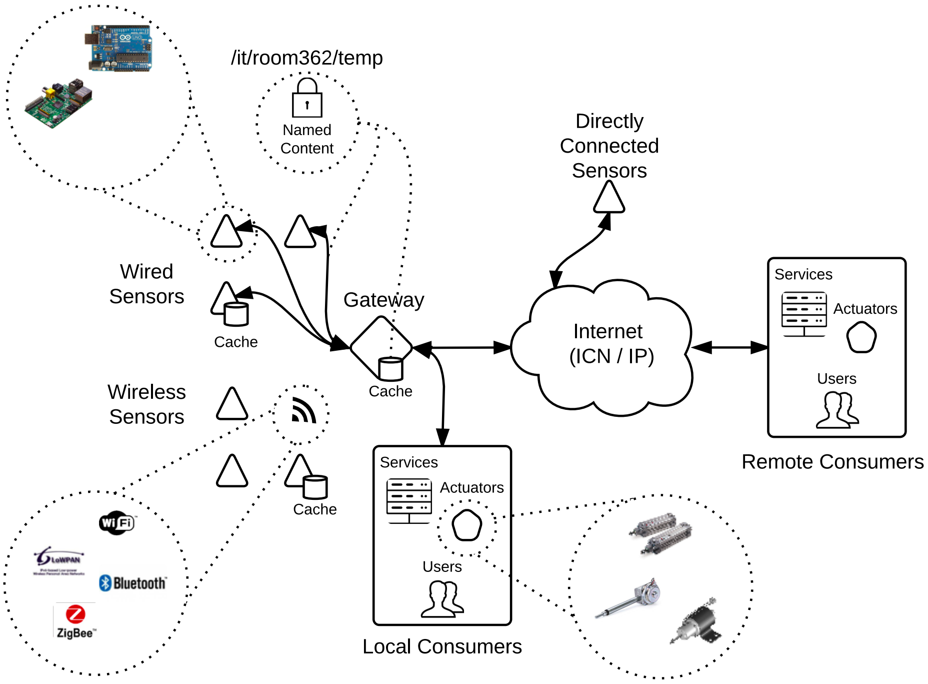


Figure 3.5: A vision on the perspectives of ICN for the IoT

Two of those issues were considered to be critical and are targeted in this work, namely the need for information freshness management and to discover content and nodes in the network. The next two chapters will then focus on the discussion and assessment of solutions to these issues.

CHAPTER 4

Keeping the Named Internet of Things Fresh

"Time abides long enough for those who make use of it."
— Leonardo da Vinci

The previous chapter highlighted that caching should be carefully handled in IoT applications taking into account data freshness requirements. This chapter tackles the management of the information freshness and how to retain the benefits of ICN's in-network caching while addressing this fundamental issue. In doing so, the current state of the art in both ICN and IoT technologies is pushed forward by developing novel concepts and shaping the ***"Named Internet of Things"****.*

4.1 Motivating IoT Information Freshness Management for ICN

In-network storage for caching is considered to be a common key advantage to all ICN approaches. The information made available in ICN can be retrieved from any other entity in the network holding a copy of the original packet. However, as it was previously established, it has some drawbacks when considering IoT-like applications, where the precision of the information received could be decisive for the adequate development of the consumer application.

In IoT scenarios, new information is continuously being generated, and consumers are mainly interested in the latest data. In this regard, the use of sequence numbers for naming data to be sensed does not seem to be a valid approach, as nodes joining the network at a given time and willing to get the latest information, do not know which sequence number to request. However, the lack of any differentiating component in an IoT naming scheme, when considering mutable content, implies that different information, with different data values, will be generated under the same name. For example, a given piece of content identified by "`/sensorX/latestTemperature`" could be cached in routers, although in fact, the stored piece of content refers to old data which will be sent back to potential consumers and will not be updated. In this way, unwanted behaviours may occur, such as the first router (or the consumer's cache itself) always replying to the consumer with old information instead of successfully forwarding interests towards the actual information source. To address this issue, NDN includes a freshness parameter (FreshnessSeconds) in Data packets (Content Objects) that establishes how many seconds a specific packet will be allowed to be held in the network Content Stores (i.e., before marking it as stale), thus giving some control over the packets removal process from network to the information producers.

However, the consumers are not able to get new content unless in accordance with the producer specified freshness and/or with the cache replacement policies of the routers. This issue gains relevance due to the fact that different consuming applications may have different information freshness requirements for the same source of information.

Consequently, in order to provide support to applications involving consumers with different freshness requirements, it seems appropriate to set the freshness parameter to the lowest value among those required by consumers. However, lower freshness values will cause the information to be retrieved from the source more frequently (i.e., fewer cache hits), leading to low efficiency in cases where the active cycle of applications with more strict freshness requirements are lower than that of the application that can handle higher freshness values, or where applications are not running at the same time. Moreover, it should be noticed that it is not always possible to know beforehand the freshness that will be required by any individual consuming application. The establishment of an appropriate value of freshness has particular importance when considering battery powered producers with constrained resources, as it is commonly the case of IoT sensors, where establishing each communication has a cost in terms of battery life.

4.2 A Consumer-Driven Freshness Approach

In this context, a consumer-driven freshness approach is proposed. In such a proposal, consumers may specify their particular requirements in terms of information freshness. This parameter will be interpreted by Content Stores, which will consider it to determine whether or not they have suitable information available. When the data with the desired freshness is not available in the cache, it will cause a Cache Miss, and the request will be forwarded towards the information source or another entity able to satisfy the desired freshness. The freshness from the producer side must be kept to ensure the appropriate control over the generated information. However, based on the freshness being requested by the consumers, the producers may take a decision about the ideal value of freshness to be included in the Data packet.

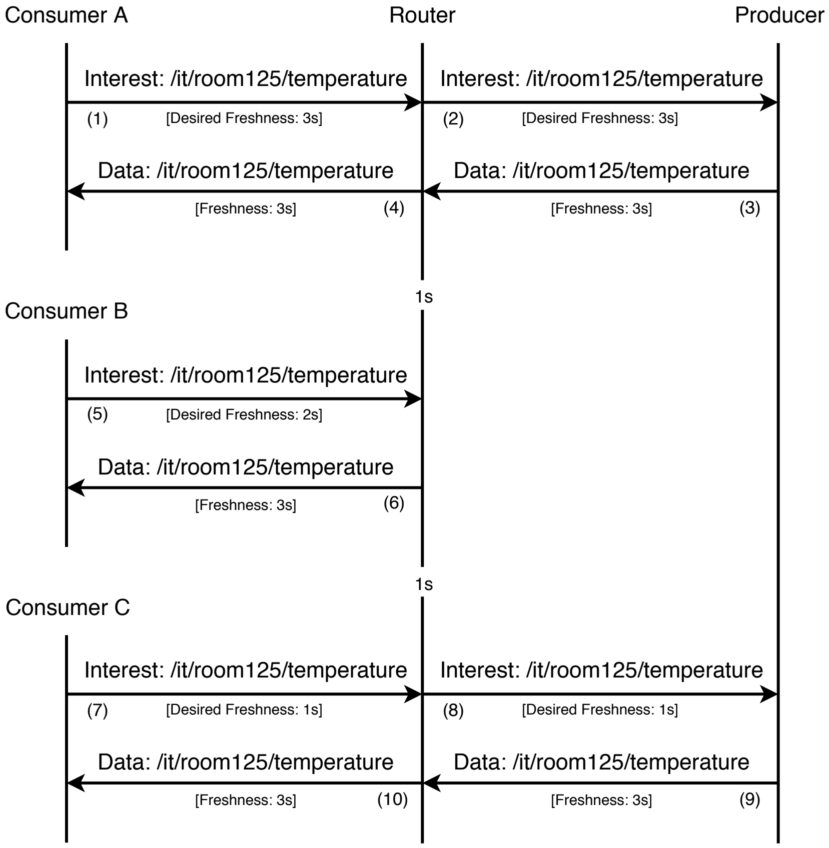

Consumer A
Router
Producer
Interest: /it/room125/temperature
(1) [Desired Freshness: 3s]
Interest: /it/room125/temperature
(2) [Desired Freshness: 3s]
Data: /it/room125/temperature
[Freshness: 3s] (4)
Data: /it/room125/temperature
[Freshness: 3s] (3)
1s
Consumer B
Interest: /it/room125/temperature
(5) [Desired Freshness: 2s]
Data: /it/room125/temperature
[Freshness: 3s] (6)
1s
Consumer C
Interest: /it/room125/temperature
(7) [Desired Freshness: 1s]
Interest: /it/room125/temperature
(8) [Desired Freshness: 1s]
Data: /it/room125/temperature
[Freshness: 3s] (10)
Data: /it/room125/temperature
[Freshness: 3s] (9)

Figure 4.1: Understanding the consumer-driven freshness approach

The operation of the mechanism proposed above is illustrated through an example in Figure 4.1. Three consumers with different freshness requirements are featured in the example. Consumer A requests the content `/it/room125/temperature` (1). Since the content is not available in the Router's Content Store, the request is forwarded towards the source of information (2). Based on this request, the producer decides to update the Data freshness value to 3 seconds and replies with the desired content (3). The Router forwards the response towards the consumer (4). One second later, Consumer B requests the same piece of content (5). Since the information is in the Router's Content Store, and it satisfies both the freshness values defined by the producer and the consumer, the existing copy is still valid and is forwarded back to the consumer (6). One second later, Consumer C requests the same information (7). Although the Router still has a copy of the information, which satisfies the freshness value enforced by the producer, it is not sufficiently fresh to satisfy the needs of the consumer, and the request is then forwarded towards the source (8). The producer then leaves the freshness value unchanged (i.e., the content could be cached for a longer time, consequently saving networking resources) and replies with the updated content (9). The content is then finally forwarded to the consumer (10).

The protection of systems against potential security threats (e.g., Denial of Service (DoS) attacks resulting from the systematic requests of low freshness values) should be considered when applying the new consumer-driven freshness mechanism. Possible approaches to secure the proposed mechanism include limiting its scope and using signed information in the Interest packets. However, these security aspects are beyond the scope of the work of this thesis.

4.2.1 Implementation Details

In implementing the proposed approach, a new optional field has been included in the Interest package for setting the desired freshness for the information being requested. A new check step has also been added to the Content Stores for determining, based on the timestamp of the currently cached Data packet, whether the stored copy meets the freshness required by the consumer or not. If a cache miss is detected, the Content Store will forward the Interest packet towards the producer. All the Content Stores will, therefore, follow this process on the path to the source and, if none of them is able to satisfy the request, it will finally get to the source which will issue the new information. Moreover, as the information in the caches is updated whenever a new Data packet for the same content arrives, other potential consumers will also benefit from this process.

A dummy freshness agreement between consumers and producers has also been considered, as described in Algorithm 4.1. According to this agreement, any request involving a freshness value higher than the one currently provided by a Producer will lead to the immediate increase of the freshness requested until it reaches a top value defined by the Producer. Such approach is now possible because the consumer is able to state the freshness value it needs and, as such, no problem will be originated if, in accordance to this agreement, the packet remains in the cache longer than expected.

Algorithm 4.1: Dummy Producer's Freshness Update Algorithm

```
Input: requested
Result: Freshness value is updated
if requested > current then
    if requested >= maximum then
        current = maximum;
    else
        current = requested;
    end
end
```

4.2.2 Proposal Evaluation

The evaluation of the Consumer-Driven Freshness Approach is conducted through simulation, pursuing two main objectives: (i) the evaluation of the impact of the freshness parameter on the network performance; and (ii) the validation of the proposed freshness mechanism. As such, the specific evaluation goals outlined for the assessment of the proposed mechanisms are summarised as follows:

1. Analyse the impact of the freshness parameter included in Data packets and the size of the network Content Stores on:
    a) The reliability of the data received by the consumers.
    b) The overall network performance, in terms of cache hits and the average number of combined hops of Interest and Data packets.
2. Assess the impact of introducing the consumer-driven freshness mechanism proposed in this work.

A simulation approach was selected for evaluation purposes because of the maturity of the simulations tools, arguably surpassing that of existing solution prototypes at the time of the realisation of the assessment. Following the same reasoning of Section 3.1.2, the ndnSim simulation tool [85] was selected for conducting the experiments.

To ensure that simulations are developed in a way closer to the IoT concepts, as well as to apply the proposed consumer freshness mechanism, some modifications and extensions to the modules provided by the ndnSIM were required. The introduced changes are mainly associated with the fulfilment of the following requirements: (i) Consumers should be able to specify the desired freshness; (ii) Interest packets should carry a new optional freshness field; (iii) Intermediary routers should take into account this field while satisfying Interests; (iv) the Producer should update the content freshness of the Data packet in accordance with previously requested freshness values. For convenience, the altered modules, as well as the purpose of the introduced modifications, are summarised in Table 4.1.

Table 4.1: Altered ndnSIM modules

ndnSIM Module	**Modification purpose**
Consumer	Asks for the same content name and specifies the desired information freshness.
Content Store	Checks if a given cached Data packet satisfies the consumer freshness requirements.
Interest packet	Includes a new optional freshness field.
Producer	Changes the freshness of the Data packet following the described dummy freshness agreement algorithm.
Wire	Takes into account the new freshness parameter field in the serialisation and deserialisation of Interest packets.

The simulated topology is composed of ten producers and five consumers, as shown in Figure 4.2. The producer nodes are WiFi stations connected to an AP in infrastructure mode. The AP is connected to a router, which in turn is connected to the router associated with the consumer nodes. The producers are randomly allocated within a 40x30 meters rectangle centred on the AP. WiFi stations were set to follow a random walk movement within the boundaries of such a rectangle. The Three Log Distance and Nakagami propagation loss models provided by NS-3 were selected to take into account the losses due to variations on distance as well as the fast fading associated with wireless channels.

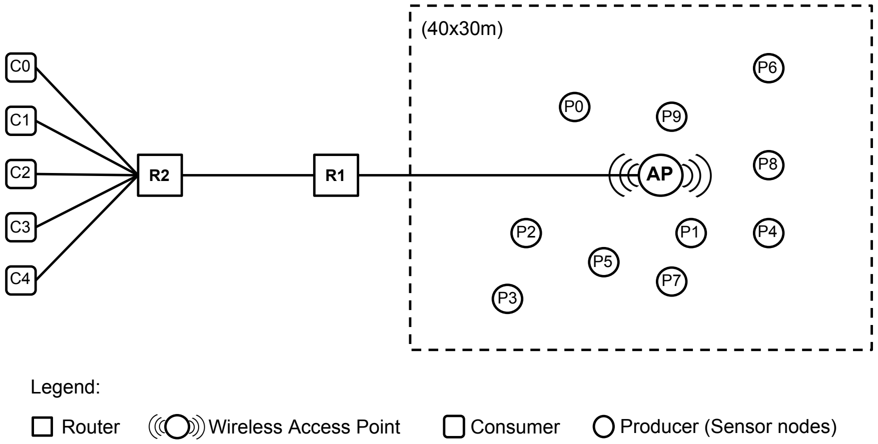


Figure 4.2: Simulated topology

For simulation purposes, it was assumed that the data sensed by the producers and being received the consumers have a sinusoidal shape characterised by a unitary amplitude and a period of 20 seconds.

4.2.3 Scenarios

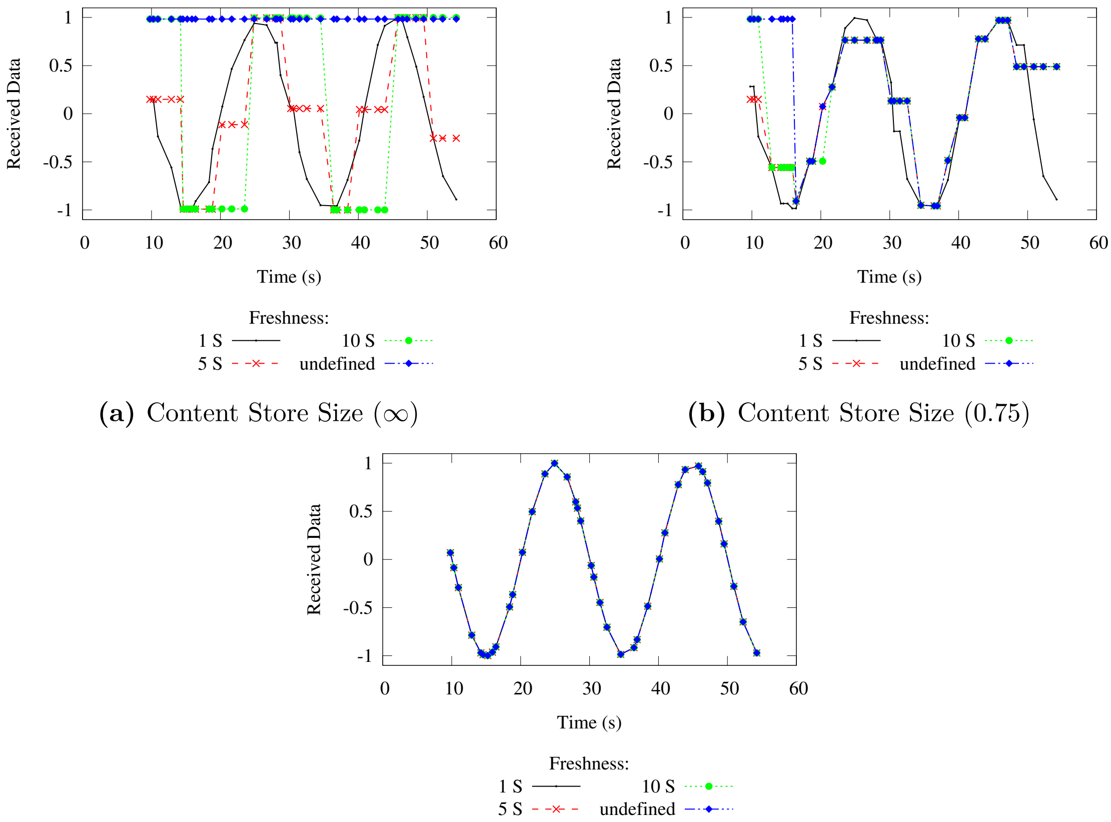
Two different scenarios are applied, specifically designed to facilitate the achievement of the targeted evaluation goals. To avoid content from being supplied by the consumers' internal cache, they were set up without cache. Additionally, consumer applications were set not to use sequence numbers but only the desired content name. In all the cases the Least Recently Used (LRU) replacement policy was considered, and ten simulation runs were conducted.

The first scenario aims at addressing the first evaluation goal. It implements the case where all the consumers have the same information freshness requirements. Consumers' applications will start and finish requesting the content at some random instant chosen uniformly within the intervals $[0; 15]$ and $[45; 60]$ seconds respectively. Each consumer will request sensed data from all the producers at a rate of one Interest packet per second with random requesting times uniformly distributed. Freshness and Content Store size are the parameters to be varied among simulations.

The second scenario was conceived for addressing the second evaluation goal. It implements a case where consumers can be divided into two groups with different information freshness requirements. Consequently, it will involve two different requirements from the network. In doing so, the percentage of active time for consumers with more strict requirements will be considered to be lower. Four of the consumers have more relaxed freshness requirements and will start and finish requesting the content at some random instant, chosen, as before, uniformly within the intervals $[0; 15]$ and $[45; 60]$ seconds respectively. In this scenario, each of these consumer nodes will now request sensed data to all the producers at a rate of one Interest packet every four seconds with random requesting times uniformly distributed and will express higher freshness values, with a tolerance for data freshness of 4 seconds. The fifth consumer will simulate a more strict freshness requirement but will be active only a fraction of the operational time of the other four consumers. This fraction is represented by the "α" parameter. Additionally, this fifth node will be considered to have a more intensive request frequency of one Interest per second and a tolerance for a data freshness of 1 second. The freshness defined by the producers was set to 1 second, to ensure the adequacy of the data received by the more restrictive node. The maximum freshness allowed by the producer was set to 10 seconds. The "α" factor and the Content Store size are the parameters to be varied among simulations.

4.2.4 Results and discussion

To address the evaluation goal 1-a) from Section 4.2.2 the first scenario was considered, and results are shown in Figure 4.3 which represents the data as received by consumer nodes for three different decreasing content store sizes, while considering the same four different values of freshness. Content Store sizes vary from a 10% (0.1) store capacity to an unlimited one. Freshness values range from the requesting Interest packet period (1 second) to its complete absence.



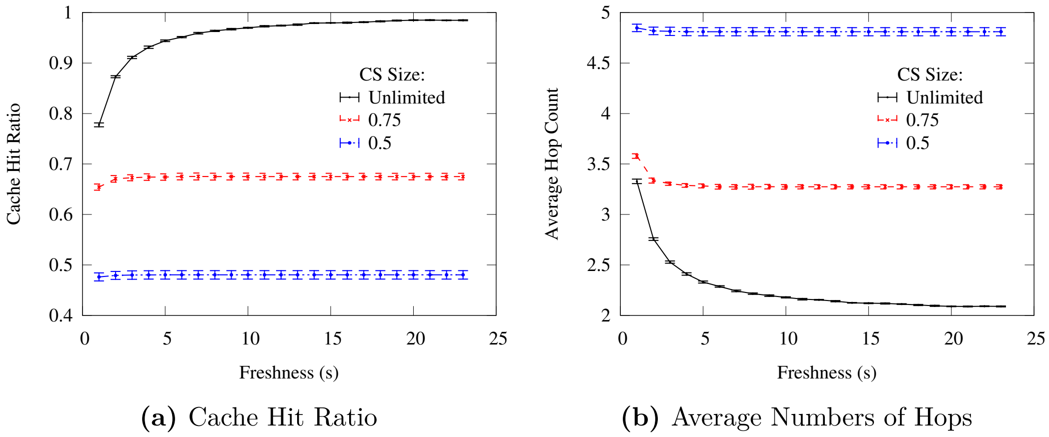
(a) Content Store Size (∞)

(b) Content Store Size (0.75)

(c) Content Store Size (0.1)

Figure 4.3: Received data for different freshness values and content store sizes

Results from this simulation confirm the benefits derived from the adoption of a freshness parameter in terms of the reliability of data received by the consumers. An analysis of the shape of the different curves Figures 4.3a and 4.3b confirm that lower freshness values lead to better data quality because the system will need to drop the stored content more frequently to satisfy the freshness established by the producers. Figure 4.3c shows that the freshness parameter loses importance as the cache capacity gets more limited because the cache will be forced to replace its content more frequently to account for the lack of storage space.



(a) Cache Hit Ratio

(b) Average Numbers of Hops

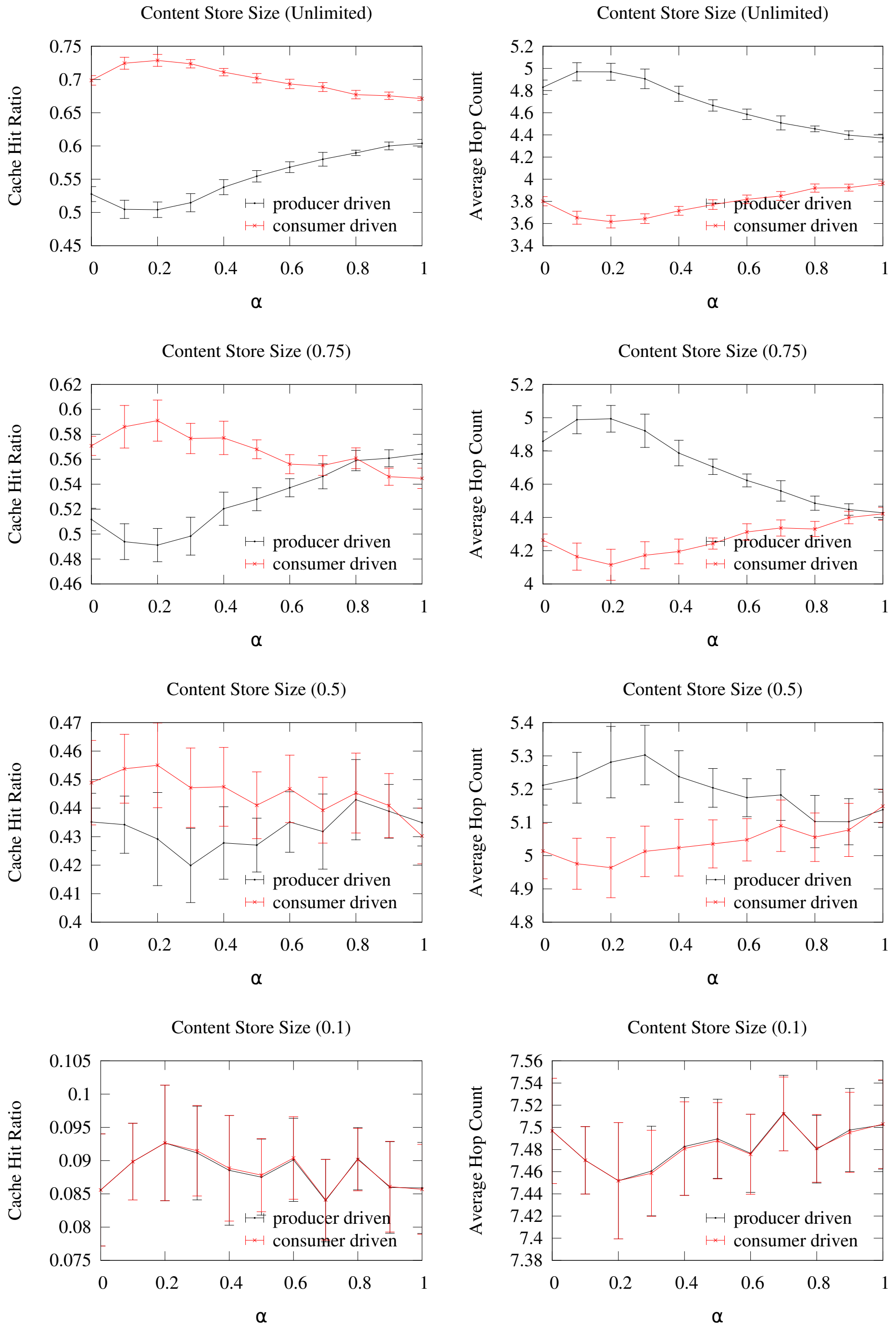
Figure 4.4: Simulation results for different values of freshness and content store sizes

An analysis of the whole picture, Figures 4.3a-4.3c reflects that, when the freshness parameter is not defined, the quality of the data received by the consumers entirely depends on how frequent the cache is forced to replace content. Simulation results highlighted the importance of the appropriate selection of the freshness parameter in accordance with the information being sensed and the requirements of the consuming application.

The first simulation scenario was also considered to address the evaluation goal 2-a) from Section 4.2.2. Figures 4.4a and 4.4b show, respectively, the cache hit ratio and the average number of hops as a function of the freshness value for different Content Store sizes. The 95% confidence interval for the data, calculated based on the result of the different runs, is also shown in both figures. Results obtained for Content Store sizes of 10% store capacity are not shown in this document, because they exhibited almost constant values regardless of the freshness value applied (cache hit ratio = 0.1 and average number of hops = 7.4 of a possible maximum of 8).

This second set of results demonstrate that some restriction should apply in adopting low values of freshness. Figure 4.4 demonstrates that low values of freshness affect the performance of the network in terms of cache hits and the average number of hops required for a given application request to be fulfilled. This is an important fact that needs to be considered in scenarios involving applications with different freshness requirements, where in order to ensure the adequate work of the system, the producer must set the freshness parameter to the minimum value among those involved in the application, thus limiting the benefits associated with ICN's in-network caching.

These results encouraged the simulation of the second scenario targeting the second evaluation goal from Section 4.2.2. That is, evaluating the effect of the proposed consumer-driven information freshness approach on the performance of the network. Simulations were conducted using both the consumer-driven freshness mechanism and



(a) Cache Hit Ratio **(b)** Average Numbers of Hops

Figure 4.5: Simulation results for different values of “α” and content store sizes for both producer driven and consumer-driven freshness mechanisms

the NDN intrinsic freshness mechanism driven by the producers. Figures 4.5a and 4.5b show the cache hit ratio and the average number of hops as a function of the Content Store sizes and “α” (i.e., the ratio among the operational time of the more restrictive and less restrictive freshness requirements of the different consumers). Figures also show the 95% confidence interval of the obtained data, calculated based on the result of the different runs.

Results in Figure 4.5 show that the use of the proposed consumer freshness mechanism led to better network performances. Namely, it can be seen that the use of the consumer-driven mechanism outperformed its counterpart by achieving, in general, higher cache hit ratios (Figure 4.5a) and required a lower average number of hops (Figure 4.5b). This difference is more evident when the Consumer with the more strict freshness requirement was active 20% of the time the rest of the consumers were active ($\alpha = 0.2$), and gets reduced as α increases. Also, the boost in performance is reduced as the Content Stores get smaller, being critical in minimal content store scenarios (Content Store Size of 10% of the total produced content), where both mechanisms lose relevance as almost no caching can be performed.

4.3 Secure Freshness Service Level Agreement

The previous section presented a consumer-driven freshness approach, in which clients can specify in Interest packets their *information freshness* requirements, thus allowing Content Stores to determine the suitability of the stored information. The current section presents a mechanism that goes one step further, and instead of having clients requesting arbitrary freshness values on every request, creates the idea of *freshness* as a service level parameter that can be negotiated and later enforced by information producers. This novel agreement mechanism is enforced as part of a fully-fledged secure IoT management architecture based on ICN presented in [23] and described below.

4.3.1 Secure ICN-based IoT Management Architecture

This section describes the information-centric IoT management architecture leveraged for the enforcement of a Freshness-based Service Level Agreement (F-SLA). The architecture is outlined in Figure 4.6. The fundamental component of the architecture is the gateway, which acts as an intermediary element in the communication between clients and IoT devices. In IoT solutions, gateways are often used to enable the interoperability between smaller device networks (i.e. Personal Area Networks) and larger-scale networks, including the Internet [100]. In this architecture, gateways also provide supporting mechanisms to connected devices in the form of discovery, registration, security, management, policy-enforcement and aggregation of content from different physical devices. These mechanisms provide a unified operating platform, capable of easily integrating a large number of heterogeneous devices and enable their secure operation. In doing so, the proposed gateway exchanges management information with IoT devices through the F_d reference point, which supports all the secure information-centric communications that are needed to execute commands and retrieve information items within the IoT deployment.

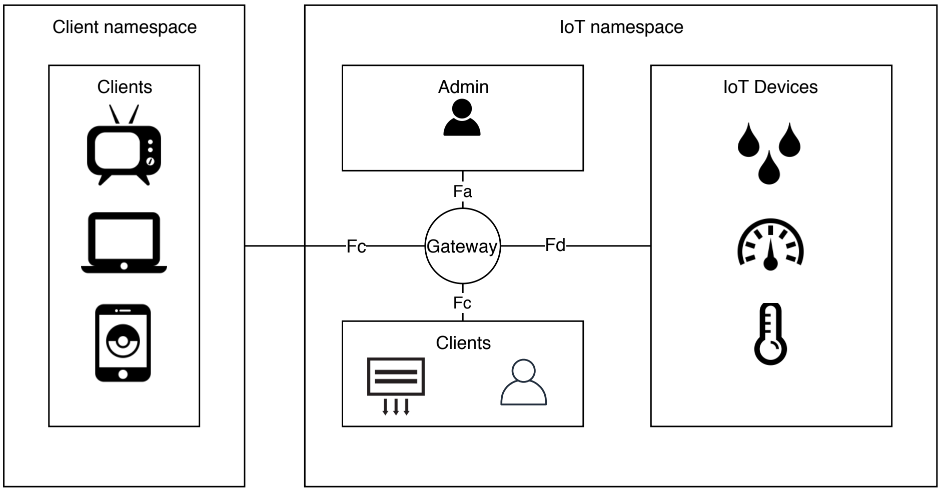


Figure 4.6: Secure ICN-based IoT Management Architecture Overview

From the clients perspective, the content-centric IoT architecture defines a set of mechanisms to support their access to IoT devices. This access is securely granted through a gateway using the reference point F_c, as shown in Figure 4.6. This reference point allows an authorised client to discover the set of IoT devices that operate under a given namespace and to retrieve a client-specific policy (previously configured by the administrators via the management reference point F_a), indicating the list of commands and information items for which that client has authorised access in the IoT devices. This facilitates to enforce the client-specific policies and prevent unauthorised clients from accessing restricted content from IoT devices, through authentication and encryption mechanisms that are executed at the gateway.

Finally, this solution can benefit from the ICN's inherent caching capabilities to allow any node between a client and an IoT device, holding a copy of content, to return the information item requested by the client. This enables a reduction of the traffic load received by the device, and thus improve the performance of IoT deployments. In particular, traffic reduction may strength scalability in terms of supported clients, while improving the energy efficiency of limited-capacity IoT devices.

4.3.2 Secure Freshness Negotiation Mechanism

The current section explores the potential of the management architecture introduced above in supporting an Information Freshness – Service Level Agreement (F-SLA). In such a mechanism, clients may send requests to the gateway for specific producers to lower their original freshness value to a given percentage *Freshness Coefficient* (i.e., $freshness_{desired} = freshness_{original} \times freshness_{coefficient}$). As such, different Service

Levels can be defined, in accordance with the freshness coefficient allowance (i.e., the minimum percentage that may be requested by a given client). The proposed mechanism considers the communication signalling described in Figure 4.7 and assumes that the gateway runs the Freshness Control Algorithm detailed in Algorithm 4.2. The proposed information exchange is based on the same primitives of the base mechanisms of the management architecture (e.g., naming conventions, signed interests, encrypted payloads) and freshness negotiation is handled via commands as defined by this architecture. As such, the fundamentals of these security aspects (e.g., authentication, key exchange procedures, encryption algorithms) are beyond the scope of this document but are provided in [23].

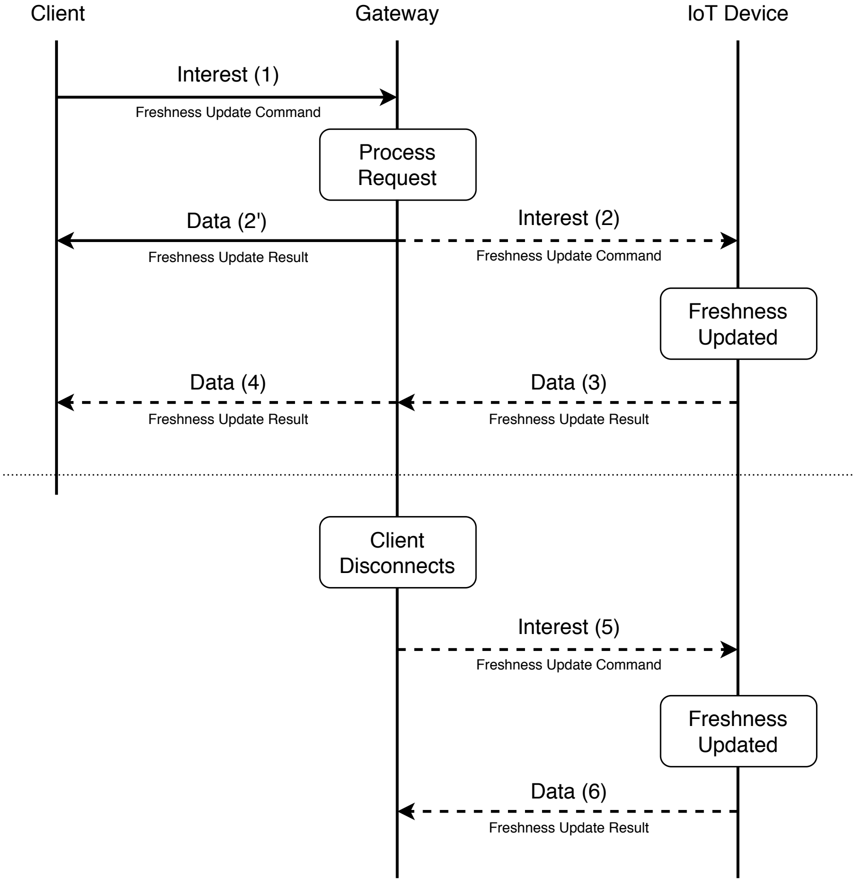


Figure 4.7: Freshness Application Signalling

As shown in Figure 4.7, a client sends a freshness update request, *Interest (1)*, specifying the desired Freshness Coefficient. Upon its reception, the gateway executes the first case of Algorithm 4.2 and adds/updates a record of this request. Requests under the minimum allowed coefficient for its level of service are set to this minimum value. Once a record is added/updated the gateway recalculates the freshness coefficient to be configured on the device (e.g., the minimum of all recorded requests). If the freshness coefficient to be set on the device does not change, the gateway replies to the

Algorithm 4.2: Freshness Management Algorithm

```
Input: New event
Result: Event is processed and the corresponding action is taken
switch Type of event do
    case Command from client do
        Coefficient = Max(Requested, MinimumAllowedByClientSLA);
        Add/Update resulting device-client freshness coefficient;
        Recalculate device coefficient;
        if newCoefficient == oldCoefficient then
            Reply to the client;
        else
            Send freshness update command to the device;
        end
    end
    case Client disconnection do
        Remove device-client freshness coefficient;
        Recalculate device coefficient;
        if newCoefficient == oldCoefficient then
            Do nothing;
        else
            Send freshness update command to the device;
        end
    end
end
```

client with *Data (2')* notifying the current freshness coefficient. Otherwise, it sends a command to the affected device, *Interest (2)*, to configure the new coefficient; the device will then send an update result notification to the gateway, *Data (3)*, which will forward this notification to the client, *Data (4)*.

When a client disconnects from the gateway (e.g., a configurable amount of time without an active session key), the gateway executes the second case of Algorithm 4.2 and the records associated to that client are removed and the affected freshness coefficients are recalculated. If there is no change in a given freshness coefficient, the gateway does nothing; otherwise, it sends a command to the corresponding device, *Interest (5)*, to configure the new freshness coefficient; the device will then send an update result notification to the gateway, *Data (6)*.

4.3.3 Proposal Evaluation

The main goal in the evaluation of this proposal is to analyse the convenience of implementing the concept of freshness as a service level parameter. With that purpose, the quality of the information received by different types of clients is assessed by relying on a simple topology, as shown in Figure 4.8. The topology is composed by three clients, with privileges corresponding to different levels of service in terms of freshness coefficient allowance: *Basic* (up to 80%) of, *Silver* (up to 40%) and *Gold* (up to 20%). Clients access through a gateway to a sensor with no access restrictions (i.e., a Basic Sensor).

As in previous evaluations, the ndnSIM 1.0 [85] simulation tool was used for conducting the experiments described in the present section. As such, the architecture presented in [23] and described in Section 4.3.1 was developed for this simulation tool.

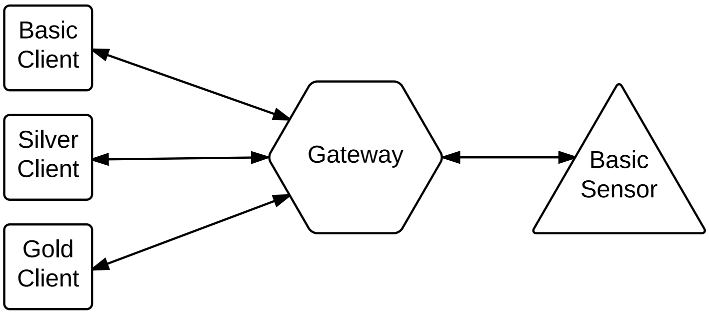


Figure 4.8: Simulated Topology

4.3.4 Scenarios

In a first scenario, the Basic client is active during the [0-100s] time interval, while the other two clients remain inactive during the whole simulation. In a second scenario, the Basic client is also active during the [0-100s] time interval, but now the Silver client is active during the [20-80s] interval, and the Gold client is active during the [40-60s] interval. Additional simulation setup parameters, selected to ensure the correct behavioural analysis of the proposed approach, are described in Table 4.2.

Table 4.2: Scenario Setup Parameters

Parameter	Value
Content Store Size	100 Entries
Client Request Rate	10 Requests/s
Requested Freshness	1%
Shared Key Timeout	60s
Device Session Key Timeout	50s
Content Payload Size	100 bytes
Device Freshness	5s
Client Session Key Timeout	10s

4.3.5 Results and discussion

For simulation purposes, it was assumed that the data sensed by the producers have a sinusoidal shape characterised by a unitary amplitude and a period of 40 seconds.

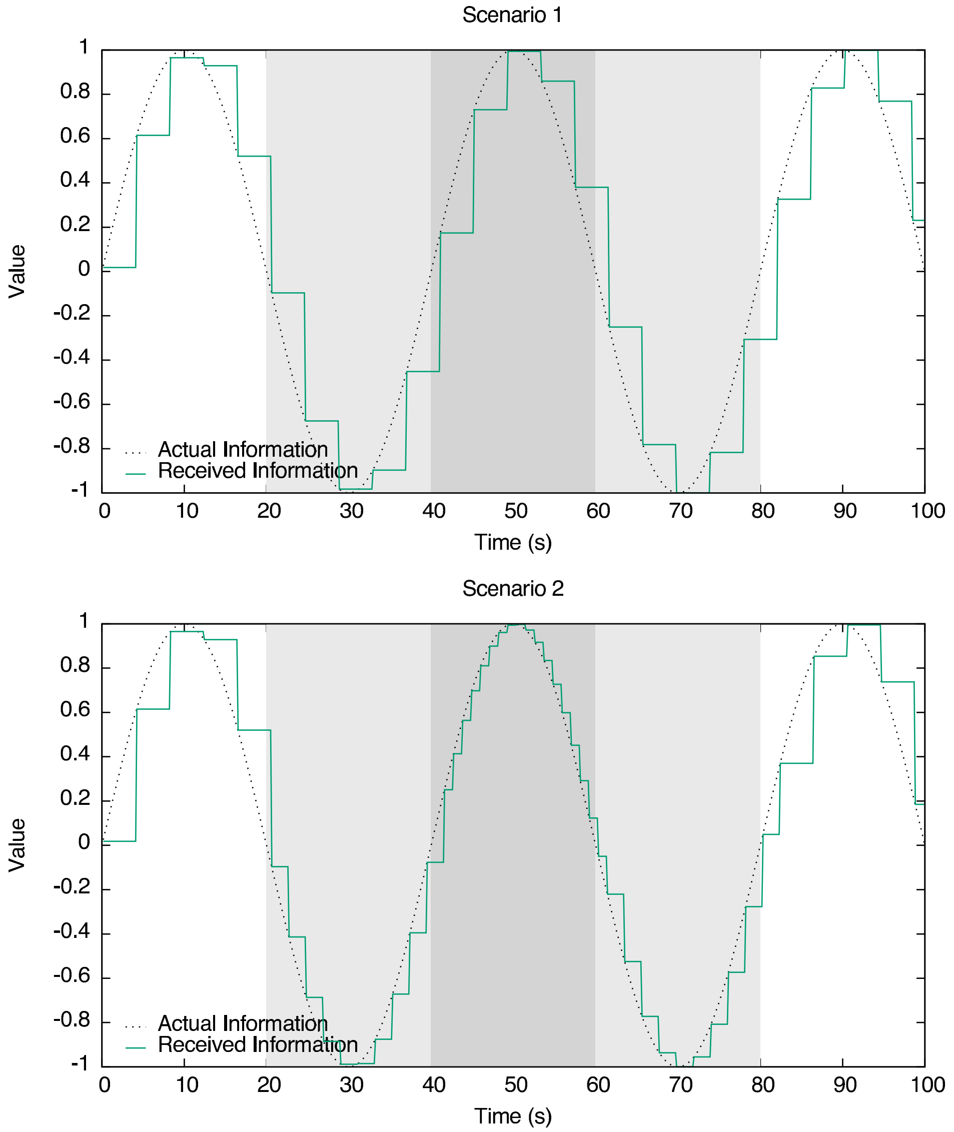


Figure 4.9: Quality of the information received by the Basic Client for the different scenarios

Figure 4.9 shows the data as received by the Basic Client. This was the selected client for this analysis, for being the one with the lowest privileges. In the second scenario, depending on which clients are active at a given time, three different situations occur during the simulation time, represented by different background colours in Figure 4.9.

A comparison of the simulation results for both scenarios shows that in the presence of the Silver and Gold clients, there is an enhancement on the quality of the received information received by the Basic Client. Concretely, the Basic client receives new information every 4, 2 and 1 seconds in the different situations, which corresponds to the 80%, 40% and 20% minimum coefficient allowance for the Basic, Silver and Gold clients respectively. Moreover, it can be observed that, as a client gets disconnected, their freshness requests expire. Consequently, upon the disconnection of more privileged clients, the quality of the information decreases. It is also evidenced that, although all the clients request a freshness coefficient of 1%, their control over the freshness value to be set at the devices is limited to the minimal allowed freshness coefficients, in accordance with the assigned privileges. However, since the information is unique, and it can be retrieved from intermediary nodes (cache), low privileged clients can benefit from the low freshness values requested by high privileged clients while active.

4.3.6 Assessing a more restrictive freshness management application

As it was already established, the previous Freshness Management mechanism is suitable in cases where the intention is to limit the resources spent in acquiring and transmitting new information and not to restrict the access to fresh information.

In this regard, a second Freshness management mechanism is proposed and evaluated. The new mechanism focuses on supporting scenarios in which less privileged clients should not get access to fresh content. As such, the gateway keeps the freshness coefficient requests in separated pools, in accordance with the type of the client issuing the request. In this way, it is possible to determine the freshness coefficient that corresponds to a given type of client. The freshness coefficient in the IoT devices must be set to the minimum among the freshness coefficients from all the active pools of clients. Then, the gateway provides copies of the content, with different FreshnessSeconds values, under different namespaces. Each namespace (e.g., `/itav/devices/device-ID/gold/...`) is tightly coupled to a client type by using different shared keys for encrypting the information, thus ensuring that each client type can only decrypt and use the content provided under its corresponding namespace.

The simulation of the second scenario of the previous subsection was repeated, and the data received by each client is represented in Figure 4.10. The results show how the usage of the new mechanism forces different clients to receive information with different qualities as defined by their SLA. Namely, the Basic, Silver and Gold clients get new information every 4, 2 and 1 seconds respectively. The notion of the different zones disappear as the freshness of the information is kept in separated pools, and the presence of one type of client does not influence the quality of the information received by the others.

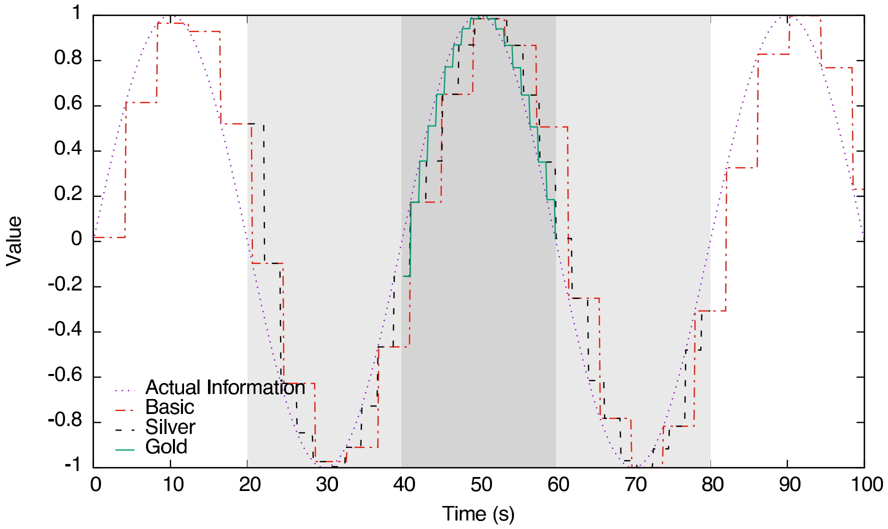


Figure 4.10: Quality of the information received by the different clients

4.4 Named Publish Subscribe Networking (NPSN)

In previous sections, the available NDN mechanism for freshness control was examined in detail, and a new consumer-driven information freshness approach for NDN was proposed. The goal of this proposal is to overcome the original mechanism limitations and provide the consumers with a way in which they may specify the freshness of the information they are willing to receive. This concept is further complemented with the notion of freshness as an SLA by integrating the freshness concept into a secure ICN-based IoT management architecture. This new concept enables the management of specific IoT content freshness according to the service level of the requesting consumer.

It is important to highlight that the FreshnessSeconds parameter is a relative value which goes signed in the Data packets and therefore cannot be modified by third parties. As such, a given ContentObject may potentially get $N * FreshnessSeconds$ old, where N represents how many times it has been cached. This issue is intrinsic to Interest-based ICN solutions, and its solution, in general, requires introducing fundamental changes to the principles of this networking architectures.

In the current section, a new strategy for dealing with the freshness of information is presented which deals with the issue in a fundamentally new way which enables the reliable retrieval of information where polling approaches are not optimal.

4.4.1 NPSN overview

The NDN protocol follows a pull-based communication approach and decouples the sender and the receiver of the information. Moreover, the sender is identified neither in requests nor in response messages. This mode of operation challenges the support for push like communications (e.g., emergency notifications) mainly in two different aspects

(i) How to push a message in a request-response communication model, and (ii) Where to address the message when the information receivers whereabouts are unknown. A simple approach is to poll producers for new information continuously. In previous sections, two different mechanisms for controlling the freshness of the information were proposed.

Notwithstanding, such mechanisms are suitable when the rate at which information is generated is known, and pull-based communication may be adequate. However, in IoT scenarios in which sensors are not always available, and communications should be minimised in order to save scarce resources (e.g., energy) this approach may not be advisable. This becomes more critical as the time at which information is produced becomes more unpredictable, and polling becomes more complex and requires a more considerable amount of resources.

In this context, a Publish/Subscribe protocol on top of NDN, inspired in the operation of the MQTT protocol is proposed and answers the previous research questions as follows:

- The producer makes a request to the consumer in order to let it know about the existence of new information (i.e., publish notification) which is followed by a request from the consumer to obtain the desired information.
- The consumer needs a routable prefix which will be used for listening to requests from the producer. The producer has to be aware of the existence of this prefix and the consumer's willingness to receive further information (i.e. subscribe).

The proposed protocol can be leveraged under two different operation modes: (i) Rendezvous-based and (ii) Standalone publisher. While the first one considers a dedicated entity responsible for matching subscriptions and publications (i.e., an operation similar to that of an MQTT broker), the second one assumes that the publisher will manage subscriptions by itself. The message sequence diagrams associated with these two modes of operation are presented in Figure 4.11 and Figure 4.12 respectively.

Independently of the operation mode, the signalling from the subscriber perspective is the same (although, for consistency, message numbering will be used as in Figure 4.12).

Each NPSN communication starts with a *connect* Interest (1) sent by the subscriber to let know the other communication endpoint (either a rendezvous or a standalone publisher) the prefix at which it can be reached. In response, it receives a Data (2) containing the *id* that will be associated with the subscriber. The reasons for using a *connect* message and generating an *id* for each client are twofold: (i) **avoiding unnecessary overhead**: each time a subscriber sends a request, it has to identify itself and relying on the full client prefix (potentially large) could incur in additional protocol overhead, and (ii) **ease the parsing of the request**: the client prefix could contain a variable amount of components, thus requiring separators for indicating the end of the client prefix, whereas using an *id* leads to a fixed length of one component. Next, it

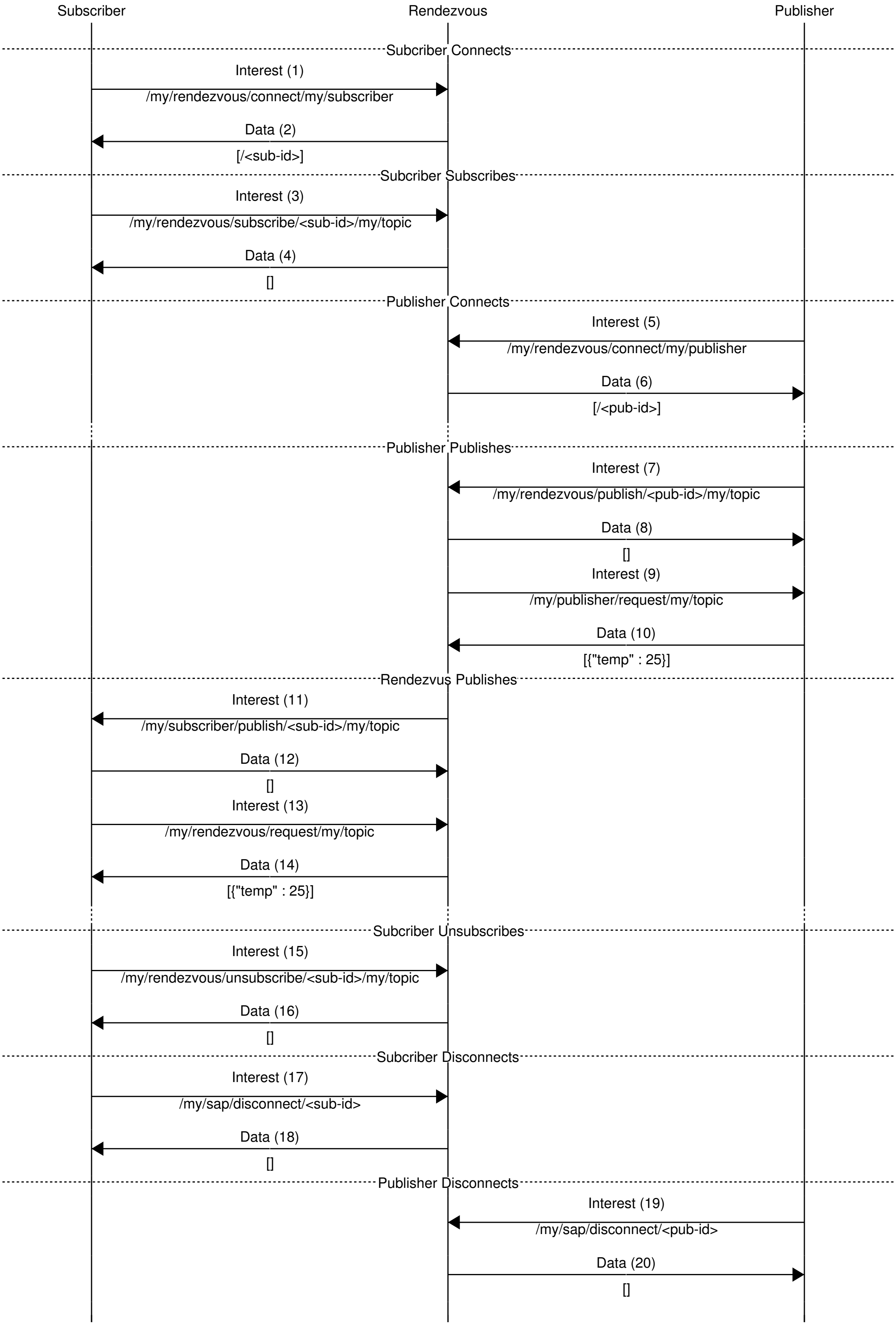


Figure 4.11: NPSN signalling for rendezvous-based operation

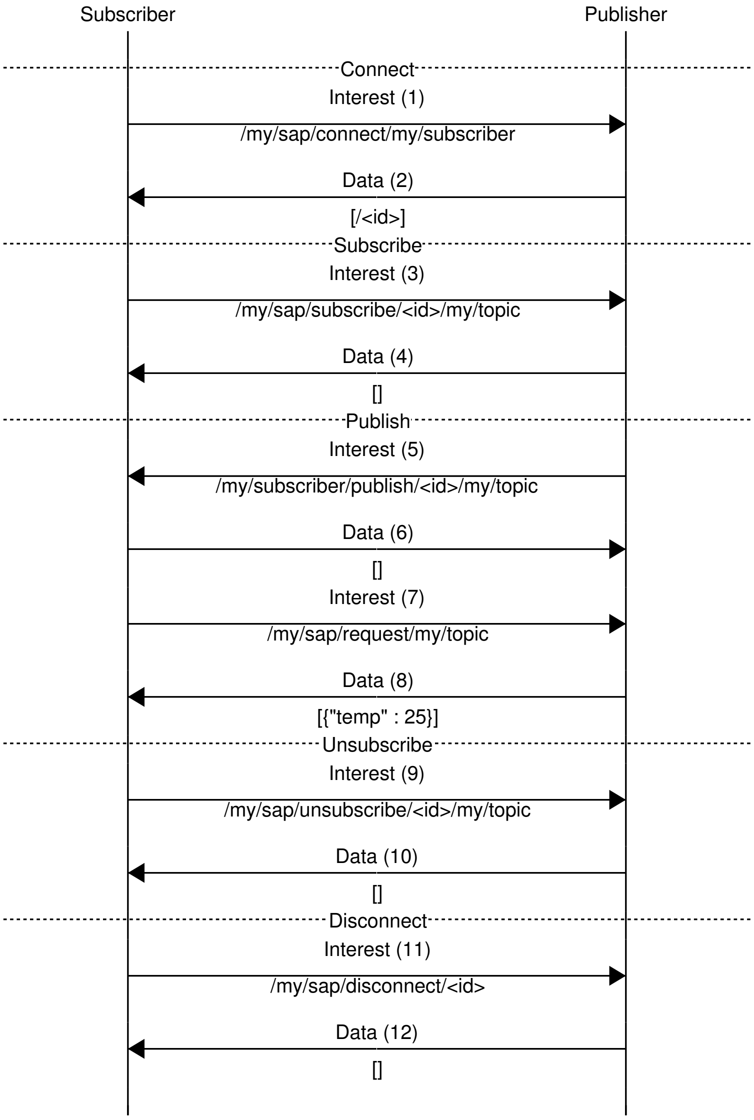


Figure 4.12: NPSN signalling for standalone publish operation

sends a subscribe Interest (3) using its *id* and expressing the topic to be subscribed. The rendezvous then replies with an empty Data (4) acknowledging the subscription (this can be extended to provide more information about the subscription process, for example, subscription lifetime). Afterwards, the subscriber awaits for *publish* Interests (5) which are acknowledged using empty Data (6), followed by a *request* Interest (7) to obtain the new information which is provided in Data (8).

The subscriber maintains the connection and remains subscribed as long as it desires to receive publications (it can subscribe more than one topic as well). When a subscriber no longer wants to remain subscribed to a topic, it can send an *unsubscribe* Interest (9) which is also acknowledged with an empty Data (10). Finally, if the subscriber wants to finalise the connection, it sends a *disconnect* Interest (11) which is also acknowledged with an empty Data (12).

It is important to highlight that the *request* Interest does not identify the subscriber and can be therefore requested by any entity on the network (even if using a regular NDN). This feature ensures that the generic advantages of NDN are maintained (e.g., those associated with caching mechanisms) as well as compatibility with NDN.

In the rendezvous-based operation mode, the standalone publisher is divided into two different network entities: (i) the **rendezvous**, which manages subscriptions and forwards publications, and (ii) the **publisher**, which sends publications. In the eyes of the subscriber, both operation modes are the same. The messages exchanged between the rendezvous and the publisher are very similar to those already described and can be observed in Figure 4.11. A publisher sends a *connect* Interest (5) towards the rendezvous which replies with a Data (6) containing the *id*. Afterwards, it sends *publish* Interests (7) to inform the rendezvous about the existence of new information, which are acknowledged with Data packets (8), and awaits for *request* Interests (9) to provide the new Data (10).

4.4.2 Discovery in NPSN

Internally a rendezvous point in this protocol already holds enough information to provide discovery over its *topics*. In this solution, the rendezvous reserves a *topic*, */SYS/discovery*, where it publishes a list of available *topics*. This suffices for providing discovery allowing potential subscribers to find available *topics*. NDN caching and naming features are leveraged to publish versioned data for this discovery *topic*. This enables a subscriber to query for versioned semantics, e.g., the latest change (`/my/rendezvous/request/SYS/discovery`), all changes since a specific version (`/my/rendezvous/request/SYS/discovery/since/%FD%05`), or all *topics* (`/my/rendezvous/request/SYS/discovery/all`).

4.4.3 Additional considerations

Although two operations modes were presented, the use of a rendezvous-based mode is advisable when publishing nodes have constrained resources. The use of a rendezvous reduces the requirements imposed over the publishing nodes, not only by reducing the amount of processing requirements associated to the management of subscriptions but more importantly by reducing the interactions with potential large amount of subscribers consumers (i.e., the publisher only communicates with rendezvous nodes instead of maintaining direct interaction with the subscribers). Moreover, the rendezvous could perform operations involving several producers such as data processing and aggregation (e.g., the average temperature in a room). In terms of security, the use of a rendezvous is also recommendable, as simpler security mechanisms may be implemented in the publisher-rendezvous interface to account for constrained devices. The way in which security for this aspect is managed is out of the scope of this thesis work, but security trust schemes for NDN are presented in [101] and particularly for the context of IoT in [23].

4.4.4 Comparing NPSN with regular NDN polling approaches

In this section, a preliminary evaluation of the NPSN protocol is conducted by comparing it with the regular NDN polling mechanism. The goal is to assess the impact of the proposed Publish/Subscribe mechanism in environments where information generation is uncertain (e.g., notification scenarios).

In doing so, information generation is modelled as a random process with a mean of 100 ms and variable standard deviation. Information is published by an NPSN publisher and consumed by either an NPSN subscriber or an NDN consumer with variable polling intervals. A total of 100 different pieces of information were generated (i.e., sensing events) and the total emulation time is considered from the time the first information is generated until the last piece of information is successfully retrieved.

From this point forward, the proposed solutions are validated based on prototyping and testing in real environments. The reasons for doing so are threefold. First, to keep the pace with the latest updates of the reference NDN solution. Second, to leverage the broad set of libraries, in various programming languages, for using this reference solution. Third, to enable the integration of the newly developed prototypes with other solutions towards the enablement of enhanced interaction scenarios.

Consequently, for evaluation purposes, a proof-of-concept prototype of the NPSN protocol and entities was developed following the NDN architecture and basing its development on the NDN Platform[1] (version 0.4.1). This scenario was deployed in an OpenStack Platform, using a virtual machine with two 3.33GHz CPU cores and 2GB of RAM.

4.4.5 Results and discussion

Figure 4.13 presents three different evaluation results which include the amount of exchanged bytes, the average delay in receiving information (i.e., amount of time elapsed since the information is generated until it is finally received at the consuming endpoint), and the percentage of all the pieces of information which were successfully gathered by the consumer endpoint. Each evaluation consisted of 10 runs and the calculated 95% confidence interval is also highlighted in the figure.

In terms of delay, it was verified that even when sampling at ten times the average production rate the delay was slightly above that obtained with the use of NPSN, while the overhead required was significantly larger, close to six times more. Moreover, while

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

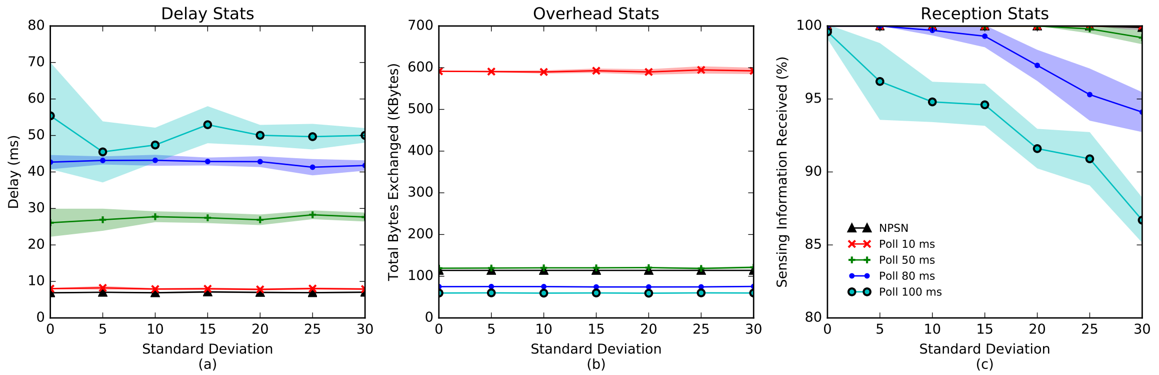


Figure 4.13: NPSN vs NDN polling evaluation results

the delay associated to NPSN is considered to be independent of the production rate (i.e., it is roughly the delay associated to the exchange of four network packets), the delay associated to the polling approach depends on the sampling rate. As such, these results are expected to be further affected as larger publications intervals are considered. Finally, from Figure 4.13, it can be also appreciated that for larger deviations (i.e., the more uncertain is the time at which information is generated) data loss may occur for low sampling rates (i.e. close to 15% of the information is lost while polling at the average information generation rate).

An analysis of the evaluation results for the NPSN protocol serves as a proof-of-concept for showcasing the benefits of using a publish-subscribe approach on top of NDN instead of relying on a simple polling based approach when the time at which information is generated gets more uncertain. However, in scenarios in which consumers have some knowledge about the time at which information is produced or in those where the impact of the delay in receiving the information is reduced, the use of polling could be more efficient. Alternatively, the use of long-lasting Interests could be explored, but attention should be paid to the usage of PIT resources.

4.5 Summary

Information freshness is considered to be a significant indicator of the quality of the received IoT information. However, keeping the information fresh while leveraging the highly distributed in-network caching provided by ICN solutions has proven to be a complex task. As such, this chapter focused on proposing and evaluating solutions towards the management of the information freshness. In doing this, in general, two different application scenarios were considered, namely those in which consumers have an insight into the information generation time, and a poll approach could be followed, and those in which new information generation is unpredictable and polling approaches are inefficient. For the first case, novel mechanisms were proposed to enable consumers to specify their requirements in terms of desired information freshness. For the second case, a publish-subscribe protocol deployable over Interest-based ICNs was proposed to ensure consumers' awareness of the availability of new information. The proposed mechanisms complement each other to account for the heterogeneity of use cases encompassed within the IoT and provide the means for ensuring that consumers can receive information as fresh as required for the particular use case.

CHAPTER 5

Discovering the Named Internet of Things

"*All truths are easy to understand once they are discovered; the point is to discover them.*"

— Galileo Galilei

This chapter establishes the importance of discovery solutions as well as the particular challenges associated with the IoT scenarios. Solutions targeting some of these challenges are then presented. First local domain discovery mechanisms are introduced for enabling discovery within the local domain when centralised dedicated entities assuming the mantel of discovery functions are not available. Afterwards, two different discovery interoperability issues are considered: (i) how to handle the heterogeneous ways of naming and structuring the information in IoT scenarios and (ii) how to discover resources which are deployed in different network architectures. The chapter finalises with an integrated view of the discovery topics presented.

5.1 A perspective on discovery topics in IoT and ICN

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 effective and efficient discovery of resources (e.g., availability, scalability, interoperability), which consequently require a high degree of automation (e.g., self-configuring, self-managing, self-optimising).

Centralised solutions ease the management of service registries, ensuring their consistency and providing fast lookup mechanisms. However, relying on decentralised 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 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.

For example, authors in [102] provide a comprehensive survey on service discovery approaches and define the prime criteria that need to be fulfilled for autonomic service discovery. Screened solutions were categorised according to (i) their level of decentralisation (i.e., centralised, distributed or decentralised), and (ii) their 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.

Also, recent research on discovery solutions for IoT environments has been focusing on the different challenges above identified. In [103], authors propose a Service Discovery solution which relies on ZeroConf [42] mechanisms and P2P technologies for integrating discovery mechanisms in both local and large scale. A fully distributed opportunistic approach is used in [104] 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 who 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 [71], the authors point out that most work related to IoT interoperability has mostly focused on resource management, and not on how to utilise the information generated. They proposed a description ontology for the IoT Domain by integrating and extending existing work in modelling concepts in IoT. In [105], 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 [106] with new mechanisms to handle service mobility in order to account for dynamic environments. A unified semantic knowledge base for IoT is presented in [107], consisting of several ontologies, namely resources, services, location, context, domain and policy. Semantic modelling is also considered in [108], 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 organise 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 [109, 110, 111], 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 [112, 113] 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 on thesaurus and Wordnet to define a semantic metric.

Service discovery topics for Interest-based ICN architectures have been also been addressed by the research community. For example, CCNx[1] (version 1.0) specifications include a proposal of a Simple Service Discovery Protocol [114] 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. While this approach is 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 beneficial for local connectivity

[1] `www.ccnx.org`

scenarios and better self-organisation capabilities.

In [66], the 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 (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).

5.2 Local Domain Discovery

Efficient device and service discovery has proved to be a complex and dynamic aspect of IoT scenarios [103]. In these scenarios, devices mainly rely on ad-hoc connections and protocols for communication; therefore, a third-party infrastructure may not always be 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 [42] use multicast/broadcast for decentralised addressing and local domain discovery. In ICN networks, service discovery is a relatively new topic, and, as referred in Section 5.1, 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 such as WiFi-Direct or Bluetooth, or in mobility scenarios where nodes cannot take for granted the availability of a dedicated discovery broker.

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).

As such, the current section targets dynamic IoT scenarios where (i) the use of brokers may not be possible, and (ii) achieving a hybrid discovery operation (both reactive and proactive) may be more suitable than a strictly reactive approach. The goal 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. In doing so, the key requirements for enabling proactive broker-less discovery over Interest-based ICNs are discussed in Section 5.2.1. These challenges are then tackled in Section 5.2.2 which introduces the concept of an alternative forwarding pipeline for local area communication to provide Interest-based ICN nodes with the capability to listen and to broadcast unsolicited ICN messages within the local network. Section 5.2.3 then formalises the proposal of a discovery mechanism, built upon the concepts of this novel pipeline, that relies on Layer 2 broadcast protocols, does not depend on a dedicated infrastructure, and supports both reactive and proactive operation modes.

5.2.1 Towards a proactive broker-less mechanism

As seen in Section 5.1, there are some approaches that enable service discovery in Interest-based architectures but they lack some important features (e.g., proactive announcement of services and support for decentralised operation) relevant in targeting IoT/M2M scenarios, where protocol efficiency is a key aspect due to the frequent 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 Chapter 2, Interest-based ICNs follow a pull-based communication model (i.e., every communication starts with an Interest packet which is consumed by a Data packet or otherwise expires). While Interest packets are routed, Data packets are forwarded based on PIT entries, which are deleted after successful forwarding. This communication model challenges the retrieval of pieces of information from multiple sources using a single Interest, which is the general case for a decentralised 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 packet the list of producers for which Data packets have been already received. The last Interest, after the content from all the producers has been already retrieved, 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; and (ii) the overhead associated with Interests is continuously increased as a new item has to be added to the *Exclude* field every time the Interest is satisfied and a new one has to be issued.

A different approach, as proposed in [74], 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. Additionally, keeping PIT entries for long periods of time could lead to PIT space exhaustion and consequently compromise the forwarding process.

(ii) Push-based communication model:

In order to enable proactive 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 [73], 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 [67], 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.

5.2.2 Elements for an alternative forwarding pipeline

The previous section identified the key requirements, as well as existing approaches for enabling a proactive broker-less mechanism in Interest-based ICNs. However, current approaches focus on forwarding issues rather than performance in constrained devices. In this context, an Alternative Forwarding Pipeline (AFP) for Interest-based ICNs could be considered to provide an additional forwarding path, based on rules other than PIT, FIB and CS matching. Such pipeline, should address the previously identified challenges, while leveraging the broadcast nature of the media. In doing so, it is expected that the proposed pipeline can provide 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.

This concept of an AFP is at the core of the discovery mechanism of the following section, where is assumed that a pipeline with such characteristics is available. The realisation and implementation of this concept are detailed in Section 5.2.5.

5.2.3 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***.

The proposal motivates the design of a hybrid (both proactive and reactive) discovery mechanism that is built upon the concept of the proposed AFP and relies on its ability to provide nodes with the functionalities of sending and receiving 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, by leveraging the novel AFP, 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 the avoidance of redundant queries by different applications, as well as ensuring a single node name.

Services are able to (un)register themselves in the local Discovery Daemon, as shown in Figure 5.1. 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 the name. The Discovery Daemon responds with a Data containing the result of the operation.

As shown in Figure 5.2, 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 bootstrap, a Discovery Daemon sends an initial query, *Interest (1)*, to check the availability of a name to be associated with the node. If the Interest times out (i.e., after a configurable amount of time), 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

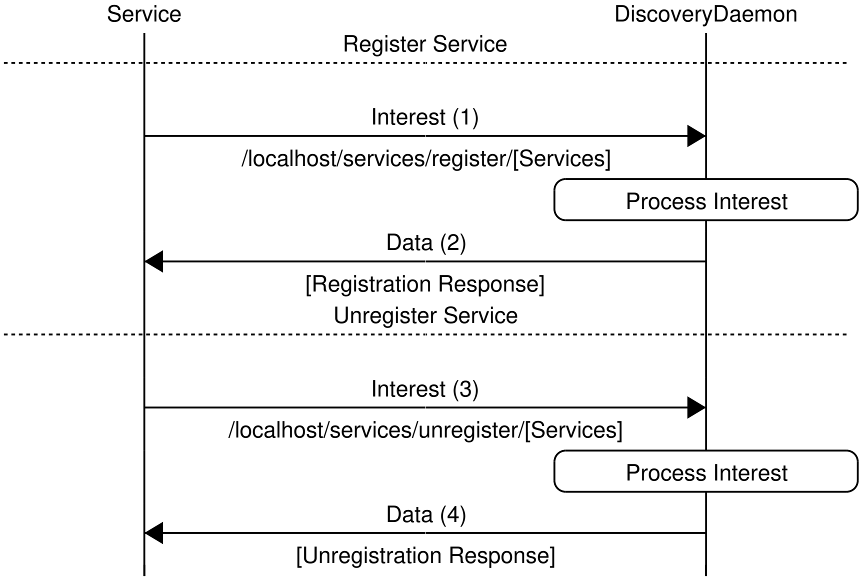


Figure 5.1: Service – Discovery Daemon Communication ***(Internal)***

in Zeroconf [42]). 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 with 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 5.3, Client applications query the local Discovery Daemon, *Interest (1)*, to find out the services offered within the local network, 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 5.2).

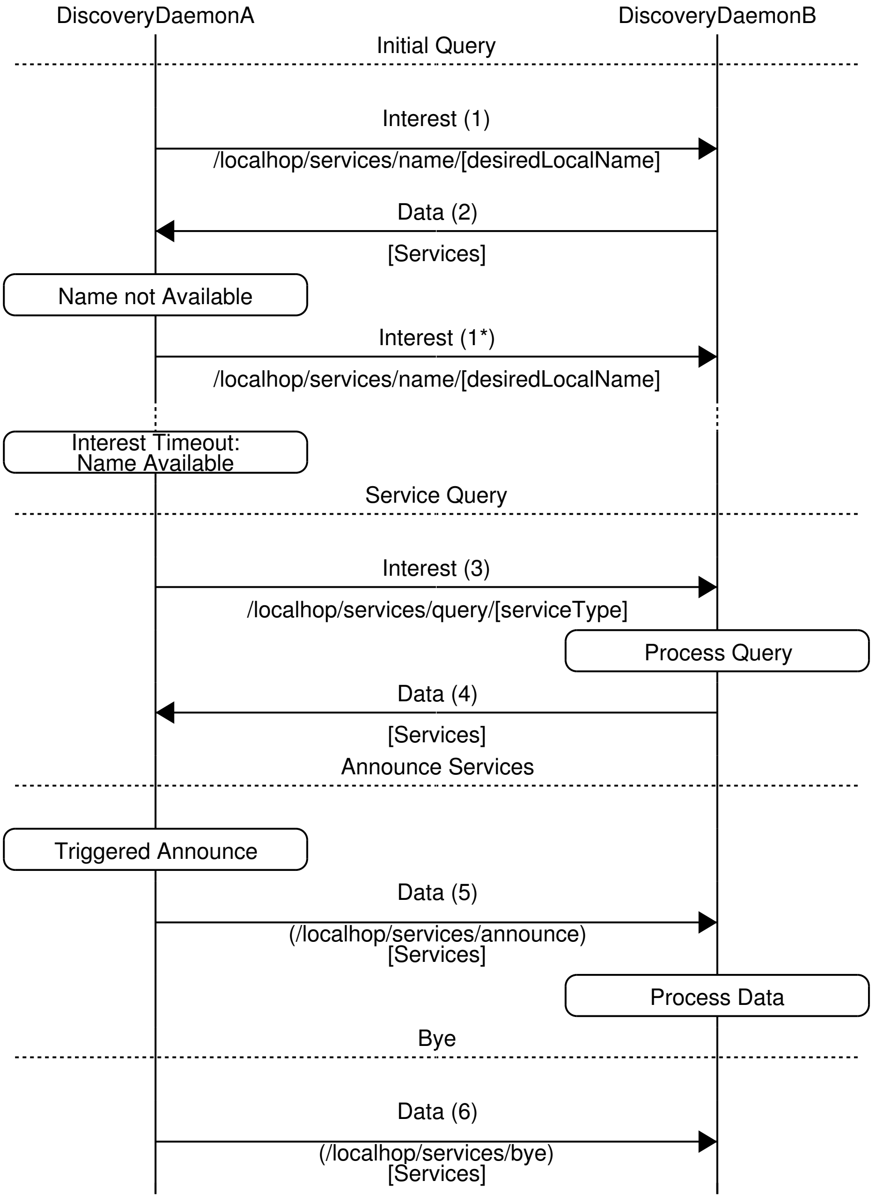


Figure 5.2: Discovery Daemon – Discovery Daemon Communication ***(External)***

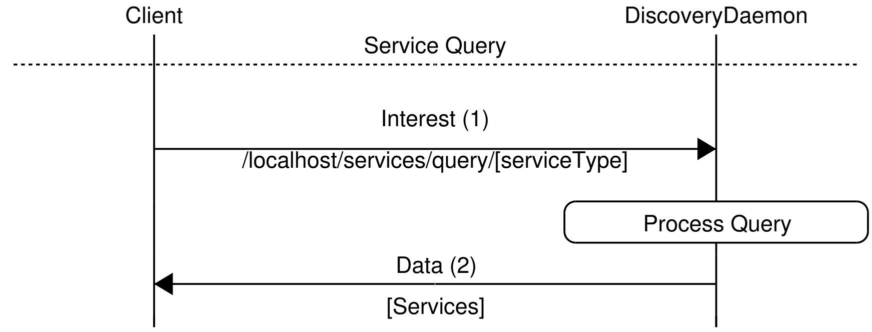


Figure 5.3: Client – Discovery Daemon Communication ***(Internal)***

5.2.4 Evaluation

The current section presents an evaluation of the local domain service discovery proposal presented in Section 5.2. The section starts by presenting the developed proof-of-concept prototype and later presents the evaluation experiments conducted by deploying the prototype in two different experimental environments: (i) Constrained and (ii) Unconstrained.

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 allow the evaluation of the behaviour of the prototype and the NDN stack in general, in the presence of devices with extremely different capabilities.

The evaluation scenario, as shown in Figure 5.4, is composed of three nodes, two service providers and one client, all of them connected to the same L2 collision domain. This setup enabled the verification of the correct operation of the prototype in multi-source environments.

In the validation of the proposal, two parameters were focused: (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.

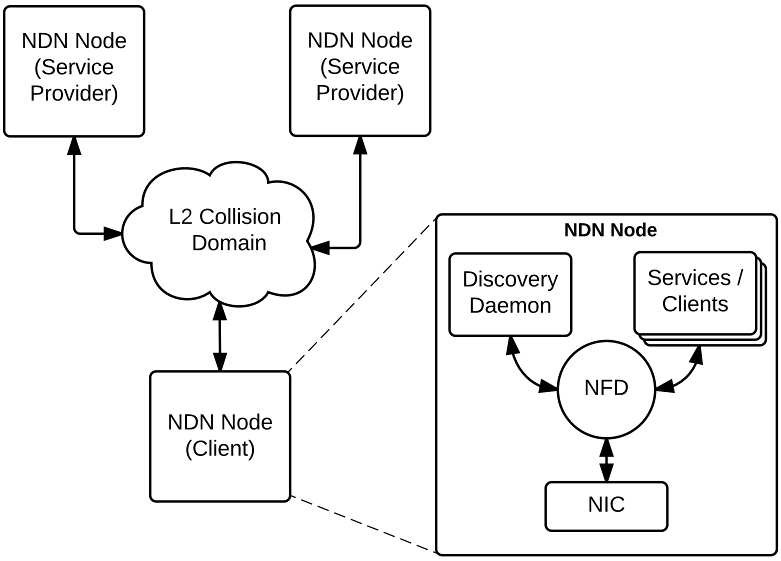


Figure 5.4: Evaluation Scenario

5.2.5 Proof-of-concept Prototype

A proof-of-concept prototype was implemented following the NDN architecture and basing its development on the NDN Platform[2] (version 0.3.2), namely on the NDN C++ library with eXperimental eXtensions (ndn-cxx) and NFD implementations. The solution is drafted on top of the NDN Platform as it is a reference open source platform for Interest-based ICNs, which is continuously updated with the latest developments of the NDN architecture, allowing the evaluation of the impact of a full NDN stack on top of both constrained and unconstrained devices. Although the solution for local domain service discovery is developed on top of NDN, the main concepts are equally applicable to similar ICN architectures.

As part of the prototype development process, besides the implementation of the Discovery Daemon, Service and Client applications, both the ndn-cxx (i.e., the reference library implementing the NDN primitives) and the NFD implementations were extended to support the novel 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 5.5 illustrates the forwarding process of an NDN node and highlights with dashed lines the extensions introduced to support the new pipeline. Any packet arriving at 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 with a 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, for the scope of the presented work, as far as it is sufficient for

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

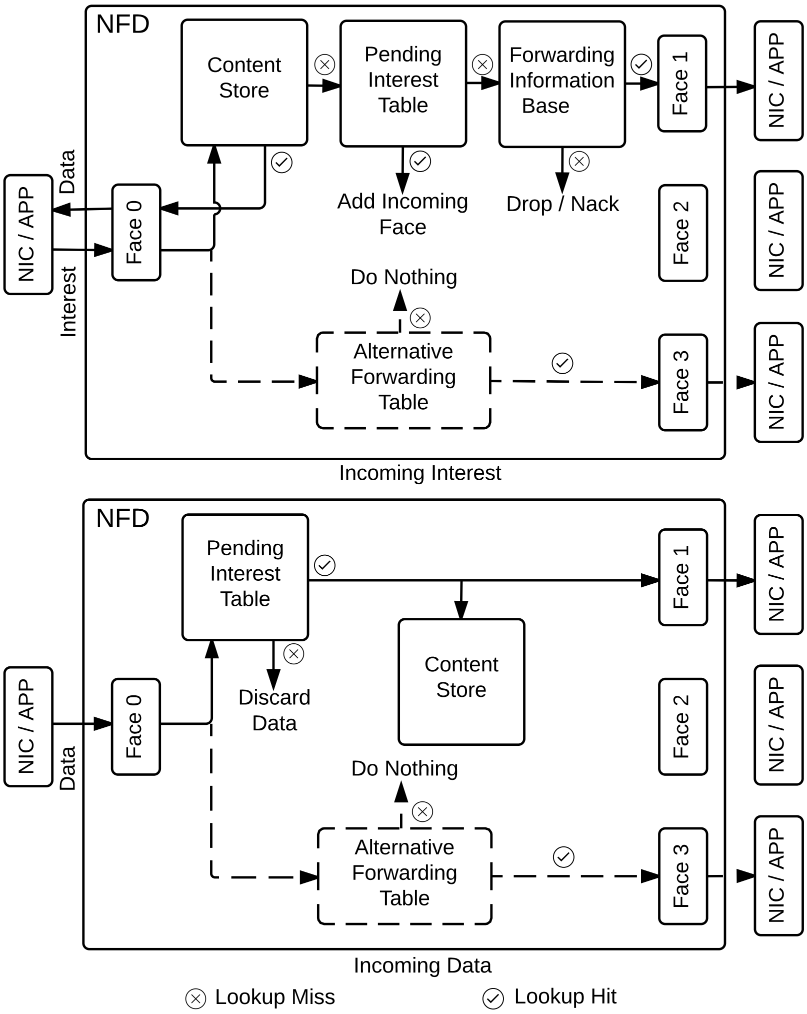


Figure 5.5: Extended Forwarding Process of an NDN Node

enabling local area service discovery, the impact of this mechanism is limited to the local domain. Also, the proposed modifications are intended only to the nodes themselves and not to the NDN routers. Therefore, the current proposal only considers 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.

5.2.6 Service time analysis

The service time was evaluated for the three main operations of the proposed discovery solution: register service, unregister service and service query (Figures 5.1, 5.2 and 5.3, respectively). 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. In order to get a better understanding of the behaviour of the proposed solution, two different approaches to request the (un)registration of services were studied:

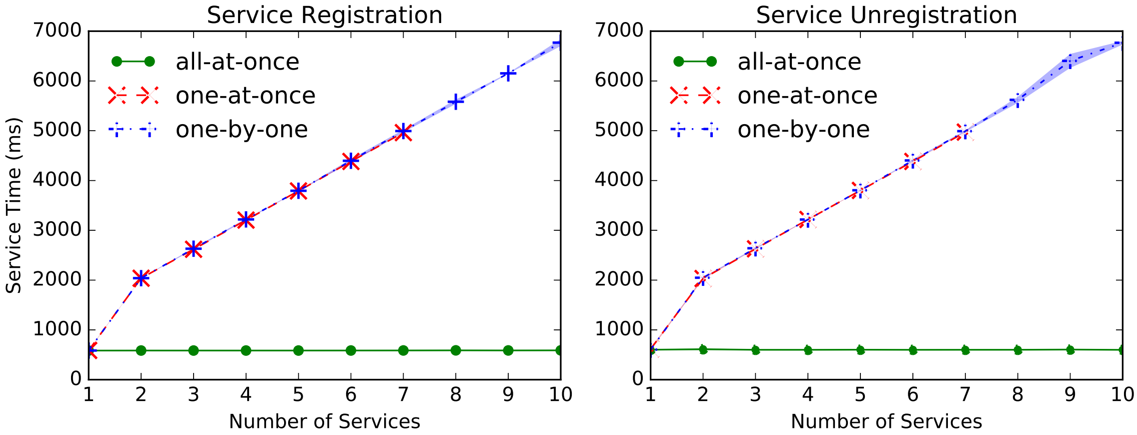


Figure 5.6: Constrained environment evaluation

(i) All services in a single request (*all-at-once*);

(ii) One service per request (*one-per-request*). This latter 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 5.6 and 5.7) for the service (un)registration 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 could not 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 in 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. 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).

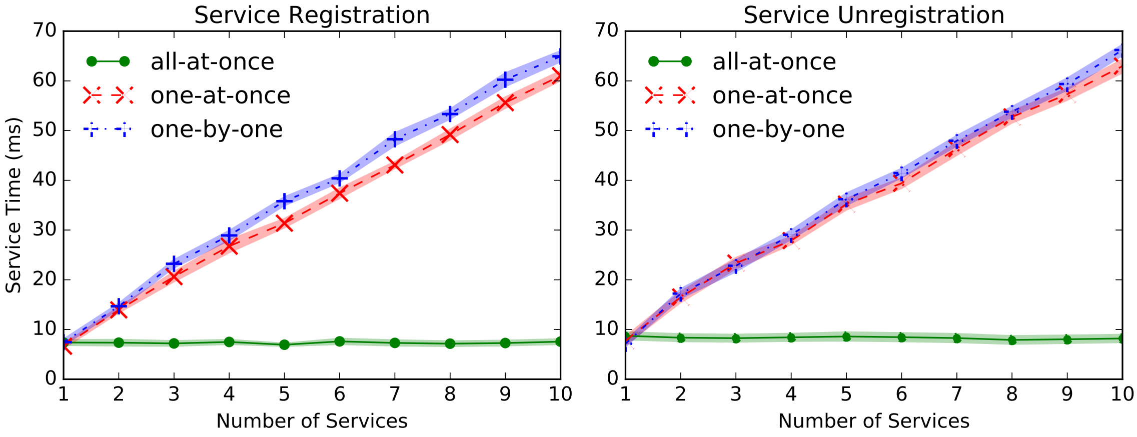


Figure 5.7: Unconstrained environment evaluation

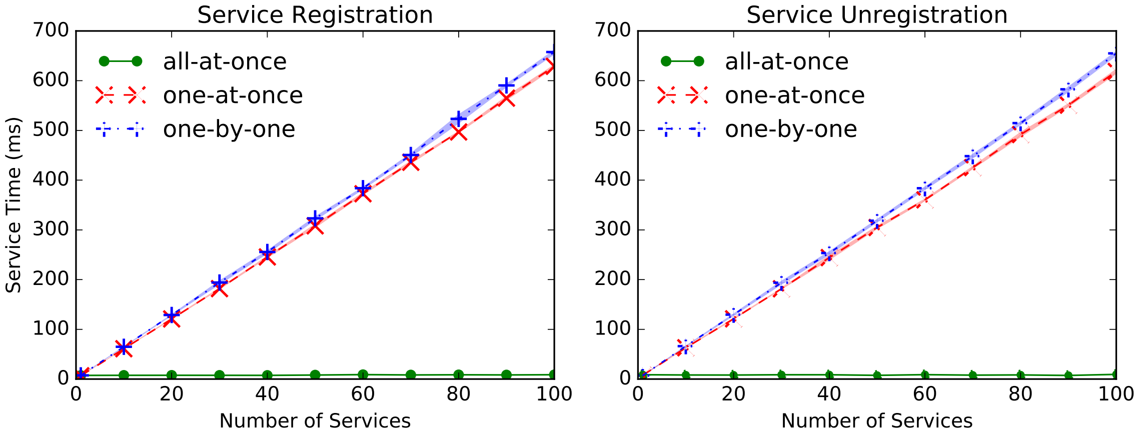


Figure 5.8: Unconstrained environment extended evaluation

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 5.8, and demonstrate that the proposed solution, when considering the *all-at-once* approach scales in terms of the 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.

5.2.7 Impact of ICN security mechanisms

The key point for variable security levels in ICN is the level of encryption 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 case in which RSA-2048 signature was used for signing data packets (default algorithm of the NDN

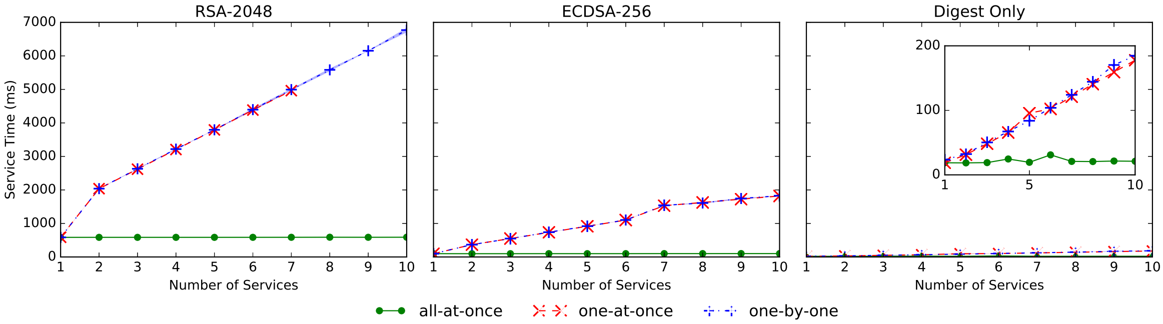


Figure 5.9: Impact of different signature algorithms in Service Registration time

Table 5.1: 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

implementation), the previous evaluation was extended for Constrained environments to study alternative signature types considered by NDN, namely ECDSA-256 and Digest Only (SHA-256). Results are shown in Figure 5.9, 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: Crypto++ v5.6.3[3] (used by ndn-cxx) and OpenSSL v1.0.1e[4]. The assessment considered the signature algorithms used by NDN, with results regarding times for signature and verification processes being shown in Table 5.1. Results show that the OpenSSL library outperforms the Crypto++ library (likely due to platform-specific optimisations), notwithstanding it remains a time-consuming process requiring further attention.

5.2.8 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 (`/localhop/services/name/nodeX`) for this implementation was determined to be 79 bytes, while the discovery Interest (`/localhop/services/query/DummyDataProvider`) was 92 bytes. As in the case of the Data packet containing the services, Figure 5.10 shows the size of the Data Packets as a function of the number of services it contains. In this

[3] `www.cryptopp.com`

[4] `www.openssl.org`

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

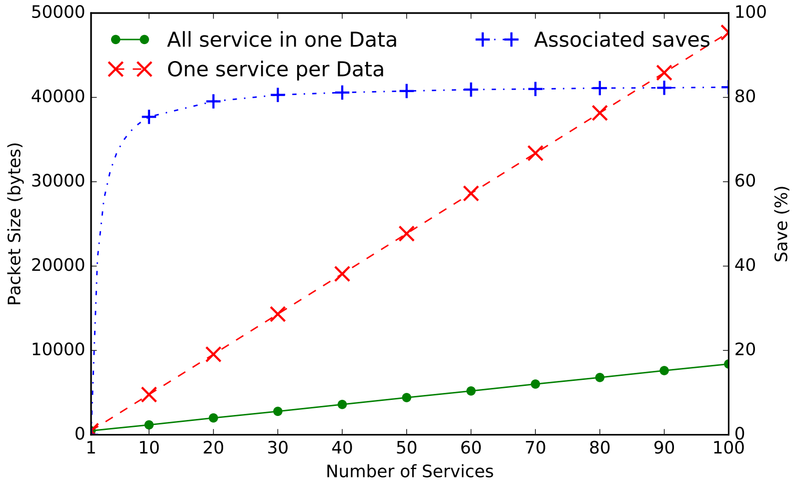


Figure 5.10: Network overhead

5.2.9 Discussion

Experimental results support the viability of using the proposed service discovery solution. Its implementation, along with the 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).

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, future deployments of this solution should explore alternative software, specifically targeting IoT devices, such as RIOT OS[5] [115], which is an operating system for IoT devices, and CCN-Lite[6], a lightweight solution compliant with different Interest-based ICN implementations.

Additionally, there is room for improvements regarding cryptographic mechanisms, namely the need for proper choices of algorithms and platform-specific optimisations. 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

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

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

hashing may be a valid approach in discovery scenarios where provenance protection is not a requirement.

Although the proposed 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 AFP 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).

Moreover, the scope of the proposed mechanisms were 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, the proposal could be integrated with other solutions (e.g., [114], [28], [66]) 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 the proposed mechanism) and to map them into globally addressable names (e.g., `/my-local-namespace/temp` maps into `/my-global-namespace/temp` and vice-versa).

5.3 Enabling Flexibility in ICN Discovery Operations

In the IoT, different devices/manufacturers specify their own structure for sharing information leading to information silos [116]. This has hindered the interoperability between different applications and the realisation of more complex IoT scenarios. Moreover, efficient device and service discovery has proven to be a complex and dynamic aspect of IoT scenarios [103]. Therefore, in order to make information useful and to ensure interoperability among different applications, it is necessary to provide data with adequate and standardised formats, models and semantic description of their content (metadata), using well-defined languages and formats [13]. 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 behind 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 [117].

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.

In this section, ICN protocols are extended with semantic discovery capabilities to contribute to its deployment and usability. In doing so, the unsupervised semantic similarity solution proposed in [88] is integrated with an ICN-based discovery mechanism developed on top of the Named Data Networking (NDN) architecture [57].

5.3.1 Solution overview

The proposed solution considers, as shown in Figure 5.11, four basic entities which interact with each other through the use of well-defined interfaces and their principal functions may be described as follows:

1) *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 communication protocol, with the Discovery Broker through the Im interface. The SME has two main functions:

    (i) Service (Un)Registering: The SME listens for requests from the Discovery Broker and accordingly adds/removes services form its local table and give the relevant feedback to the broker;

    (ii) Service Matching: The SME listens for incoming queries from the Discovery Broker, performs the matching between queries and services, and replies with a list of the relevant services (i.e. those for which there is a positive matching between the terms included in the query and the tags used to describe the service).

2) *Discovery Broker:* The entity responsible for holding the information about the available services and for matching incoming queries against the available services. This matching procedure is conducted by interacting with the Semantic Matching Engine (SME). It communicates, using the NDN protocol, with the interested Clients through the Ic interface and with the Service Providers through the Is interface. It also communicates with the SME either over the NDN protocol or an available transport protocol (e.g., UDP, TCP) through the Im interface. Under the scope of this proposal, 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 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 work, the focus will reside on the matching capabilities provided by the SME. The functions of the Discovery Broker include:

(i) Service (Un)Registering: The Discovery Broker 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 SME in order to keep updated the services database located at the matching engine;

(ii) Service Matching: The Discovery Broker listens for discovery queries from clients, forwards them to the SME and based on its response, answers to the client with a list of the matching services.

3) *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: Clients issue a request to the Discovery Broker to find out available services 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.

4) *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 Is interface, and with the interested Clients through the Ir interface. Service Providers, support two operations:

(i) Service (Un)Registering: Service Providers send 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: Service Providers listen to interests from potential clients and satisfy them by providing the corresponding content.

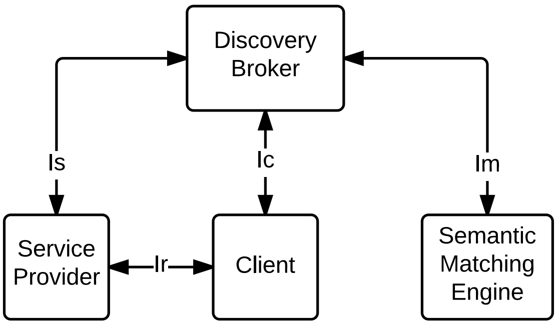


Figure 5.11: Solution overview: entities and interfaces

5.3.1.1 Detailed Communication Procedures

The procedures followed by the different entities to communicate with each other can be summarised as follows.

1) Service (Un)Registration Procedure

Services, in order to be discoverable, must be registered in the Discovery Broker, as shown in Figure 5.12. The Service Provider, sends registration interests, $Interest(1)$, to the broker responsible for the corresponding namespace. These interests contain 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 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, $Packets(5-8)$, but only the ids of the services are included on the unregistration requests.

2) Service Discovery Procedure

Clients, as shown in Figure 5.13, in order to discover the available services, must send a query, $Interest(1)$, to the Discovery Broker. Queries include a semantic description of the desired services. The broker forwards requests 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 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.

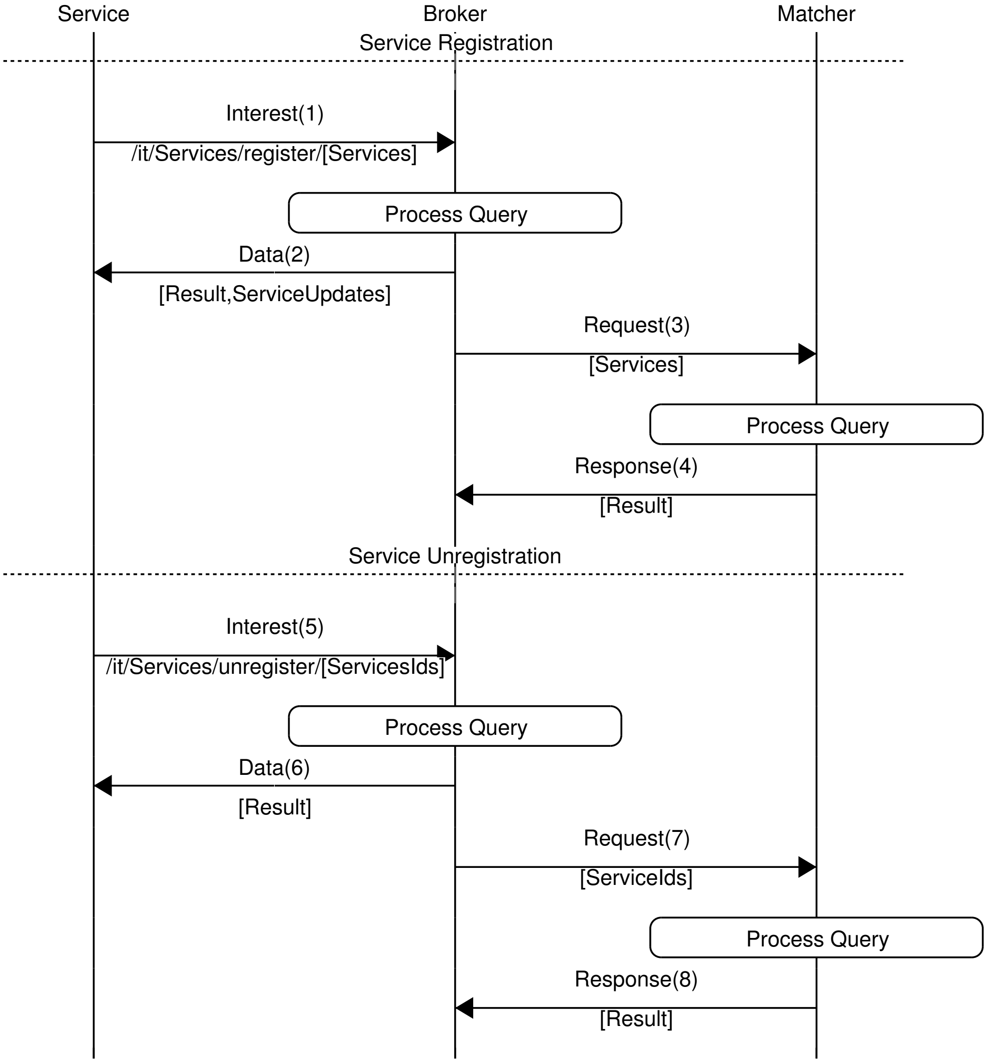


Figure 5.12: Service (un)registration message sequence diagram

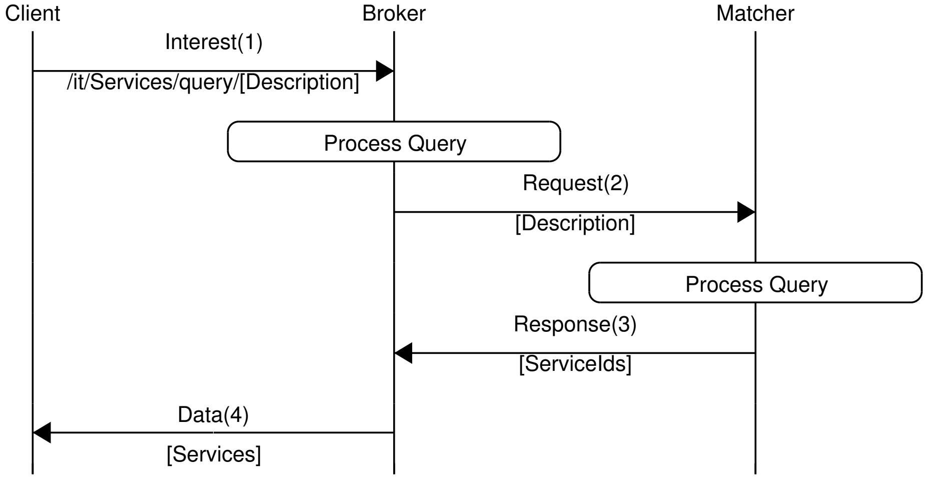


Figure 5.13: Service discovery message sequence diagram

5.3.1.2 Semantic Matching Engine: Detailed Description

During the design of this mechanism, the core concepts of the solution proposed in [88] were extended with novel functionalities for supporting service discovery mechanisms turning it into a fully-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 matching 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 5.14, receives two terms as input and returns the semantic similarity between them. Cosine similarity (Equation (5.1)) 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: acts as a bridge between the solution and the search engine (e.g., Bing[7] and Faroo[8] APIs);

(ii) Text Processing: a pipeline that processes and cleans the corpus;

(iii) Distributional Profile Extraction: analyses the output of the previous pipeline and extracts the profile of the term.

$$\cos(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \tag{5.1}$$

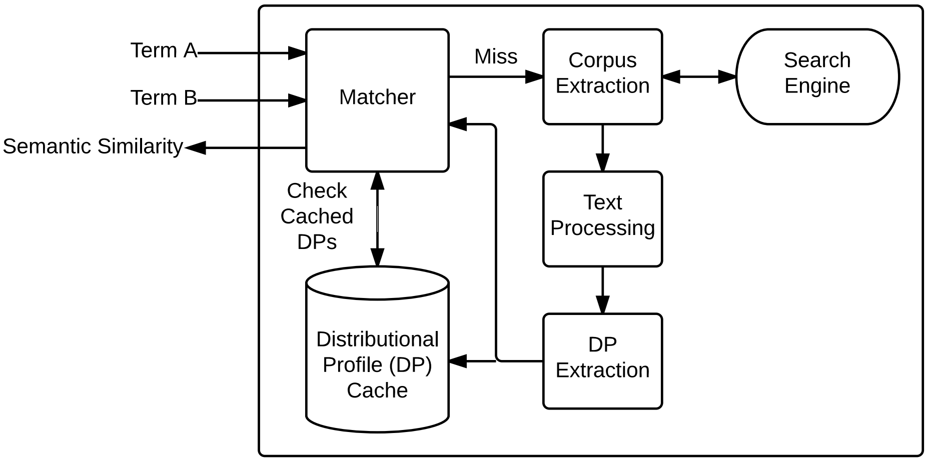


Figure 5.14: Semantic Matching Procedure

[7] www.bing.com

[8] www.faroo.com

While the original work in [88] was limited to the extraction of distributional profiles based only in unigrams, it was updated to 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 SME, 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 a set of words Jaccard Index (Equation (5.2)) and Cosine similarity are considered.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.2}$$

5.3.2 Evaluation

In this section, the proposal of Section 5.3 is evaluated by deploying a proof-of-concept prototype into an experimental environment. Similar to the previous evaluation, this proposal is assessed by focusing 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.

For evaluation purposes, a proof-of-concept prototype was developed and deployed 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, an outdoor wireless testbed composed of 24 fixed nodes focused on increased controllability and high reproducibility of the experiments [118]. Each node runs an Ubuntu 12.04 OS on top of 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. The evaluation scenario was composed by a Broker, a Semantic Matcher, a single Client and a single Server.

5.3.3 Proof-of-concept prototype

For implementing the proof-of-concept prototype, as for the case of the previous prototype (see Section 5.2.5), the NDN architecture was selected and based its development on the version 0.3.2. 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.

5.3.3.1 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[9], carriots[10]) a set of terms commonly associated with IoT services were extracted, as well as different ways of referring to those services. Using this information, a dataset was designed to adequately describe the scenarios expected to be part of the IoT (e.g., Smart Cities, Smart Agriculture, Domotic, Home Automation). As such, the dataset is composed of services and queries, each of which is described by four keywords. In the case of the queries, three different approaches were considered:

(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, eight groups of queries were defined as described in Table 5.2. The resulting dataset is composed of 30 services and 240 queries. Each service has eight queries associated, each of which falls into one of the mentioned groups.

Table 5.2: 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**

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

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

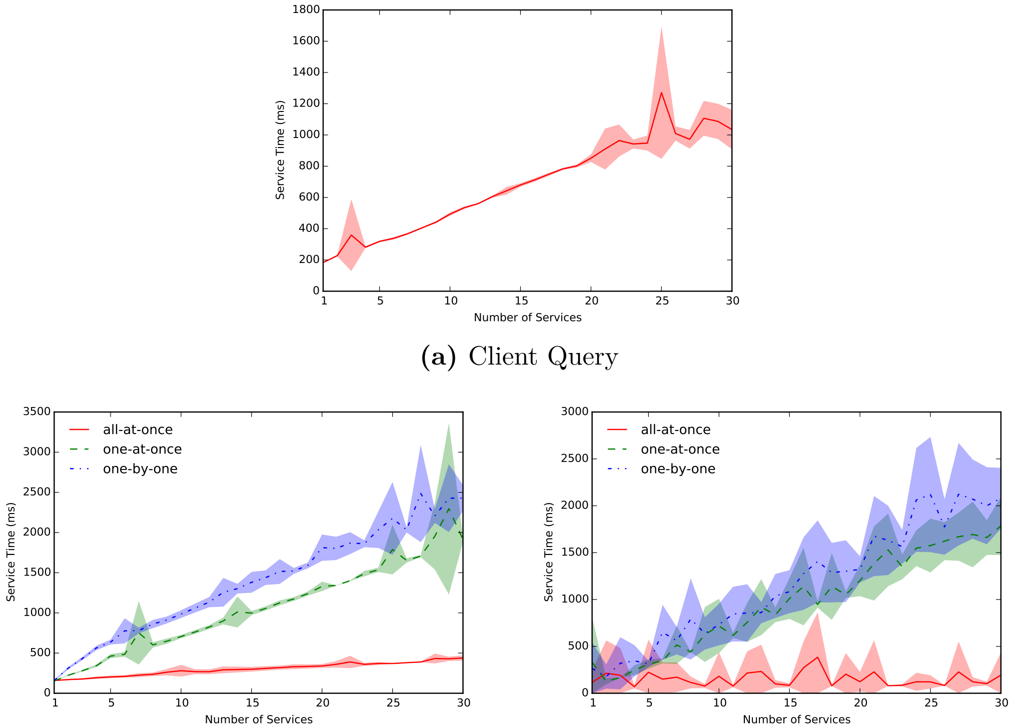
5.3.4 Service time analysis

As in the service time analysis provided in the previous section (5.2.6), the three main operations of the solution were analysed: register service, unregister service and service query (see Figures 5.12 and 5.13). In this occasion, the number of services being processed in each evaluation varied from 1 to 30 (with a resolution of 1 service), while the same two different approaches to request the (un)registration of services were studied (i.e., *all-at-once* and *one-per-request*, which is in turn subdivided into *one-by-one* and *one-at-once*). 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 5.15. Figure 5.15a shows the service time for the service discovery operation performed by the Clients. These 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. 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 5.15b and 5.15c show the results for the registration and unregistration process, respectively. Results show that the service time for unregistration procedures are shorter (approximately 500ms) 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 numeric identifier of the service.

Using the *all-at-once* approach, results show that there is not a considerable increase in 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 in the service time. The reason behind this behaviour includes the involvements of larger network overhead (as will be seen in the next section) and also due to the need of processing a larger amount of packets at the different layers of the network stack.



(a) Client Query

(b) Service Registration

(c) Service Unregistration

Figure 5.15: Service Time

5.3.5 Network overhead analysis

This section provides an analysis of the network overhead at each interface of the solution. Table 5.3 shows the results for the reference 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 with 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 with the service discovery process for the individual request and the aggregated request strategies, respectively. However, it is reasonable to assume that, in general, 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.

Table 5.3: Network Overhead

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

5.3.6 Semantic matching performance

Finally, the performance of the different string matching algorithms (i.e., exact string matching, Levenshtein distance of 2 and semantic similarity) was evaluated 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 almost identical and therefore, for the remainder of this section presents only the results obtained for the Cosine similarity.

Figure 5.16 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 (darker tones indicate higher precision). The average precision was calculated using Equation (5.3), 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 evaluations purposes, only the service associated with the query was considered to be relevant.

$$AP = \frac{\sum_{i=1}^{n}(P(k) \times rel(k))}{number\ of\ relevant\ documents} \tag{5.3}$$

Figure 5.17 represent the Mean Average Precision values in the form of a boxplot where the lines represent the 95% confidence interval for the results. Using the same representation scheme.

From Figures 5.16 and 5.17, 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 matched to the relevant service. However, the semantic similarity matching still manages to get the matching service, although not in the proper rank.

From Figure 5.18, which represents the processing time for the different matching algorithms, it can be established that the semantic matching is a time-consuming

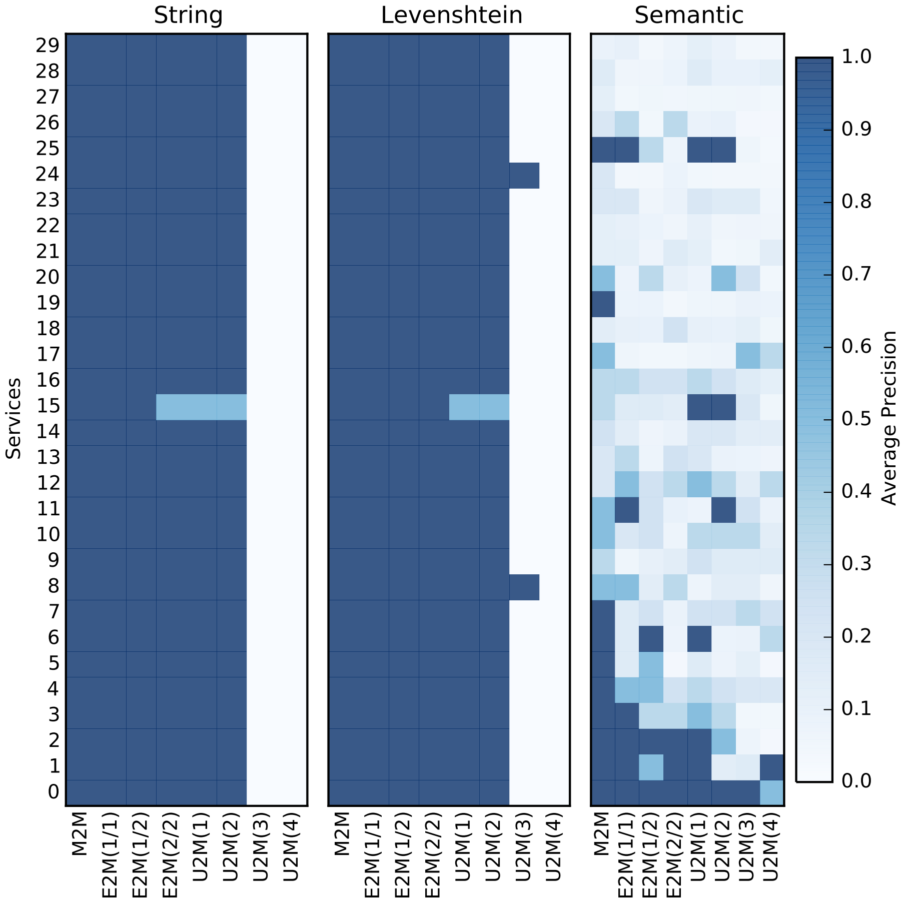


Figure 5.16: Average Precision Heatmap

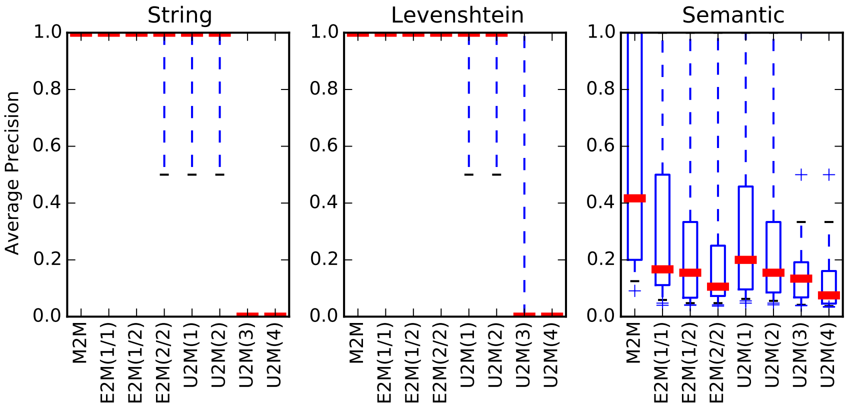


Figure 5.17: Mean Average Precision Boxplot

process, between 500 and 600 ms, thus introducing a delay in the service discovery process and therefore requiring further attention.

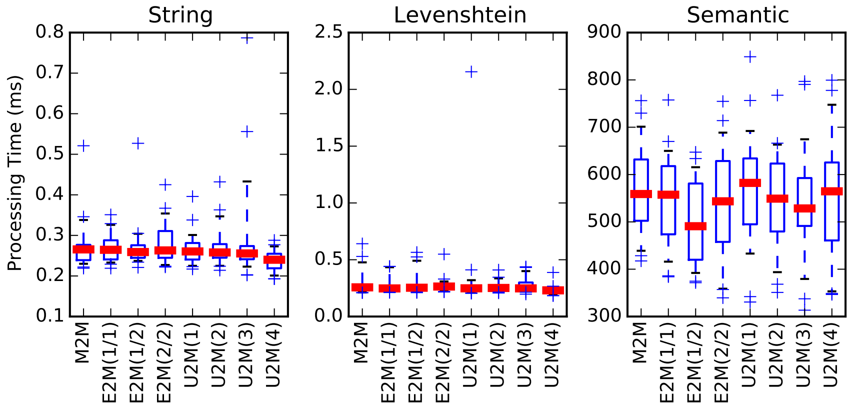


Figure 5.18: Processing Time Boxplot

5.3.7 Discussion

An overall analysis of these results (Figures 5.16–5.17) shows that the current approach constitutes an initial step into further refinements of the semantic matching algorithm. However, these also demonstrate the feasibility of using such techniques. Particularly for the case of the queries that include three and four synonyms, where the conventional methods did not obtain a match for the service, but the semantic method found some matches. The results also encourage to consider not only the individual results for one isolated mechanism, but also a weighted sum of the outputs of different mechanisms. The low performance of the semantic mechanism on the E2M groups suggests the possibility of considering words within the Levenshtein distance during the evaluation of the distributional profiles of a given term. The use of word thesaurus may also be leveraged for 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.

Moreover, the application of the semantic matching concepts into ICN scenarios should not be limited to those presented in this thesis and future works could extend the application of matching engines to the network layer itself (e.g., forwarding in meaningful namespaces, routing in flat namespaces).

5.4 Interoperable Discovery

On the previous section, IoT discovery interoperability was addressed from the information structure perspective. In this section, a different view on this interoperability matter will be presented.

Introducing new mechanisms for a better IoT at the network layer requires an adaptation/co-existence period with legacy operations such as IP. This fact is further exacerbated by the plethora of IoT scenarios, each with different requirements in terms of mechanisms used for supporting data inter-exchange. As such, an interoperability issue is raised. Setting up discovery functionalities for ICN, and establishing mechanisms for propagating this information across different network architectures are key elements in achieving interoperable IoT environments, enabling content to be mapped and made discoverable between different networking architectures by transferring the necessary operations into the network elements.

5.4.1 Interoperable IoT in ICN

The proposed solution leverages the flexible naming seen in ICN, to facilitate device discovery and corresponding mapping across conversion gateways. Conversion gateways, as the name suggests, are responsible for the conversion of both data and control information from one network architecture to another. In particular, the proposed solution will target as in the previous approaches Interest-based ICN solutions. Notably, this proposal will be based on the NDN solution. In addition, supporting push-based communications in ICN is fundamental for power efficiency, but this issue requires further improvements for IoT environments [20, 19]. As such, this proposal will mostly focus on publish-subscribe, as one way to provide push-based communication by building up on top of the previously introduced NPSN protocol (Section 4.4). In doing so, the primary concerns can be summarised as follows: (i) to advance mechanisms for IoT deployments using NDN; and (ii) to support interoperability with existing IoT infrastructure.

Lets consider Figure 5.19 as a reference example for this proposal, where two technologies interoperate: an NPSN and MQTT. A group of sensors (NPSN sensors in the figure) is organised around an NDN publish-subscribe broker (NPSN rendezvous in the figure, described in Section 4.4), and a conversion gateway (Future Internet eXchange Point (FIXP) entity in the figure, described in Section 5.4.2 and detailed in [25]) provides interoperability with the MQTT protocol in order to allow clients from the IP network to retrieve information from the sensors.

To enable this scenario, the conversion gateway must first discover resources on the NDN network, referred to as *topics*, i.e. a reference to a data source. It then advertises this information via a discovery protocol (in this case multicast DNS is used [42]). Following this setup process, a consumer can discover the availability of a *topic* and subscribe to its updates using the MQTT protocol.

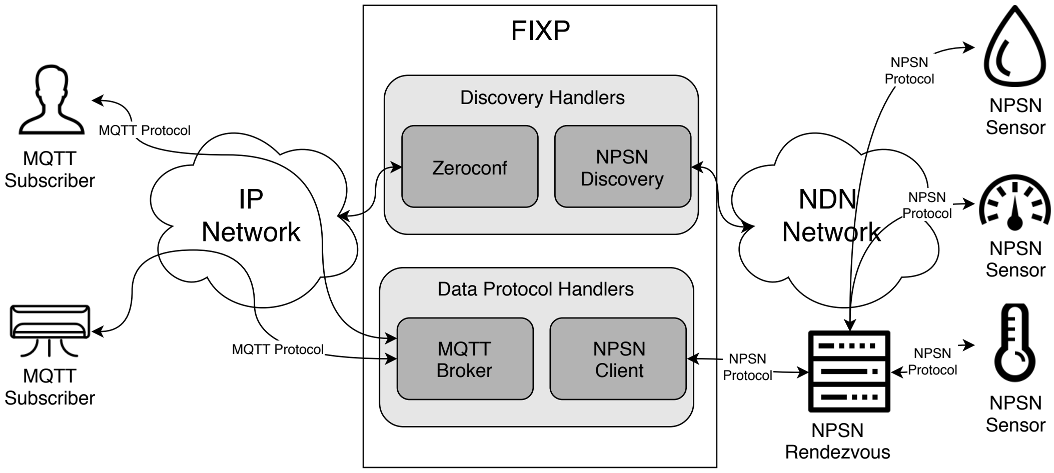


Figure 5.19: Interoperability scenario overview

5.4.2 Future Internet eXchange Point (FIXP)

To expose sensor data across protocols, a network entity responsible for message conversion, called FIXP, is considered. The FIXP behaves, on each network architecture, as the source and sink of information, converting messages across network architectures. The role of the FIXP can be split into two tasks:

1. Discover available resources in the source network, setup communication gateways for the corresponding endpoints and advertise them in the target network;
2. Process messages related to the mapped resources, such as requests and corresponding responses.

Considering the reference scenario in Figure 5.19, the FIXP acts as an MQTT broker and an NPSN client following the signalling presented in Figure 5.20 to enable the interoperable data exchange between the MQTT Subscriber and NPSN Sensor. For simplicity, it is assumed that the NPSN sensor is already connected to the NPSN rendezvous and that both the endpoints remain connected to their respective brokers.

The MQTT Subscriber starts by establishing a connection to the FIXP (1–2) and subscribing a topic via an *MQTT Subscribe* message (3). Upon the reception of this message, the FIXP acknowledges the subscription (4a); connects to the corresponding NPSN Rendezvous (4b–5); and subscribes the corresponding topic (6–7). Afterwards, every time there is a new content the NPSN sensor sends it to the NPSN Rendezvous (8–11), which in turn sends it to the FIXP (12–15) before it is finally delivered to the MQTT Subscriber (12).

While the mechanism developed for this thesis focuses on interoperable discovery, further interoperability examples, covering other protocols and conversion directions, as well as the impact of the interoperability operations on the data path, could be found in [24, 25].

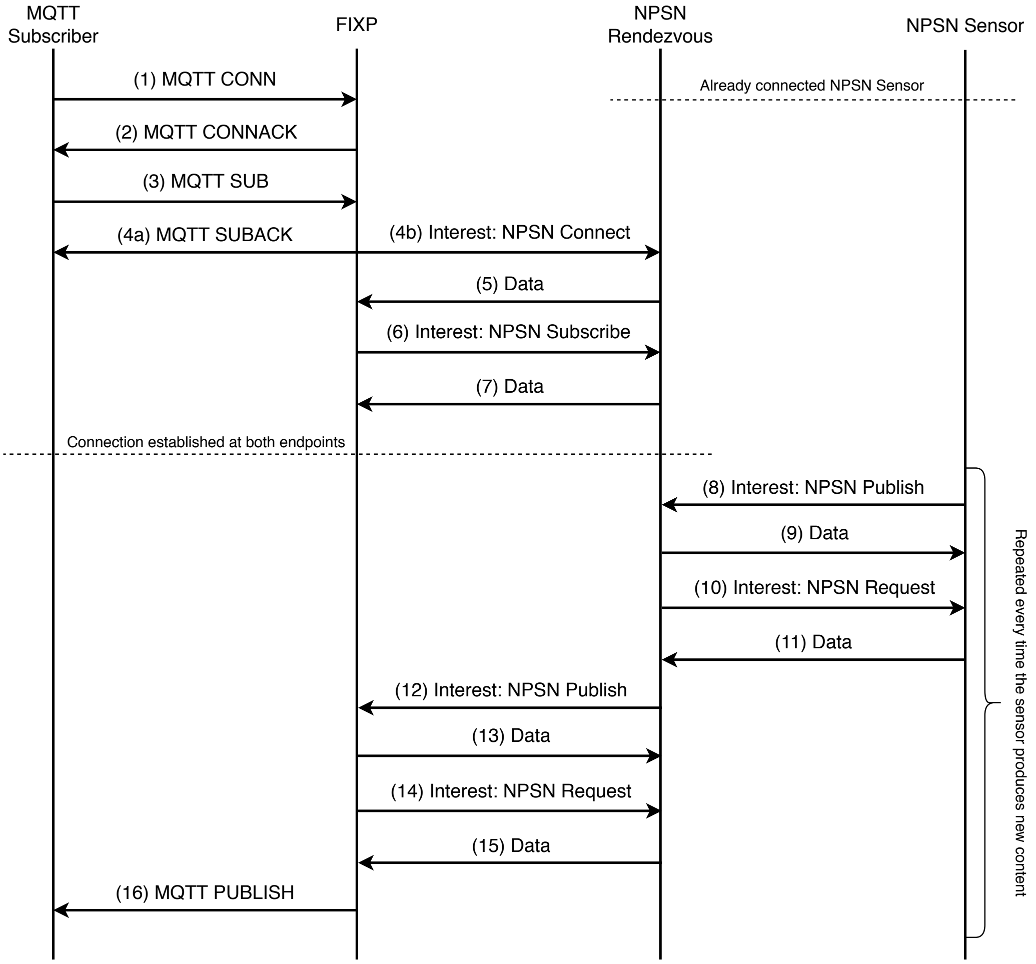

MQTT Subscriber
FIXP
NPSN Rendezvous
NPSN Sensor
Already connected NPSN Sensor
(1) MQTT CONN
(2) MQTT CONNACK
(3) MQTT SUB
(4a) MQTT SUBACK
(4b) Interest: NPSN Connect
(5) Data
(6) Interest: NPSN Subscribe
(7) Data
Connection established at both endpoints
(8) Interest: NPSN Publish
(9) Data
(10) Interest: NPSN Request
(11) Data
(12) Interest: NPSN Publish
(13) Data
(14) Interest: NPSN Request
(15) Data
(16) MQTT PUBLISH
Repeated every time the sensor produces new content

Figure 5.20: Interoperability signalling

Discovery in the source network architecture can rely on the mechanisms provided by NPSN (Section 4.4). The FIXP is a gateway that maps a source URL (NDN, in the proposed reference scenario) onto N foreign URLs (e.g., MQTT, CoAP), the URL scheme identifies the protocol used to access this resource. Since MQTT has no network discovery mechanism, or *topic* enumeration protocol extension, multicast DNS is used here to fill this gap and to advertise discovered topics in the foreign network. For consistency, resource naming follows the scheme from [119]. This holds some benefits over other types of identification (such as random, or sequential identifiers), in that it can unambiguously determine a source URL as corresponding to a foreign endpoint since each new URL always includes a unique UUID across all protocols. This facilitates gateway chaining because a gateway can always identify the original resources. The FIXP handles subsequent messages, relating to the mapped resources, in the source or destination networks. Requests for content using MQTT are translated into NDN messages, and content retrieval using NDN is translated into MQTT messages.

5.4.3 Evaluation

This section proposes an evaluation of the mechanism proposed in Section 5.4 in an interoperable IoT scenario, where IP-based MQTT devices interact with enhanced NDN publishers supporting the NPSN protocol and extended with discovery and publish-subscribe capabilities.

Following the deployment in Figure 5.19, the following metrics are considered:

1. The expected delay before a consumer can access resources discovered by the FIXP when it reacts to new devices in the source network;
2. The performance of the FIXP acting as a conversion gateway, after the previous step;

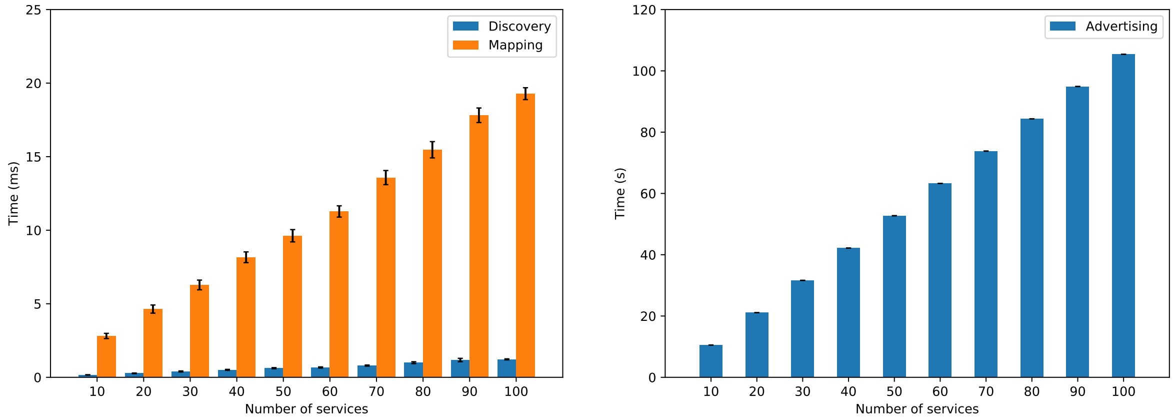
The scenario in Figure 5.19 was deployed in a virtual environment. The NPSN sensors were emulated, with a new sensor appearing every 5ms, each producing new values every 2s. The scenario was deployed in an OpenStack Platform, with each entity running on a virtual machine with two 3.33GHz CPU cores and 2GB of RAM.

A proof-of-concept prototype was developed by extending the framework proposed in [25] with NPSN and MQTT functionalities, as well as with support for discovery functionalities at the core of the operations of the FIXP.

5.4.4 Interoperable deployments operational assessment

Figure 5.21 presents the average delay introduced by the FIXP to discover new sensors and make them available in a different network architecture, for increasing sensor bursts (i.e., the number of appearing sensors in each run), with a 95% confidence interval (for 100 runs). More specifically, the delay in discovering NPSN-enabled resources in the NDN network, creating compatible URIs in MQTT and advertising them via multicast DNS in the IP network. As expected, the delay on each step of this procedure increases with the number of discovered devices. Also, the advertising delay is the step that introduces the biggest delay. It should be noted that there are limits to the ability to use multicast DNS for advertising sensor availability. The packet MTU limits the amount of advertisements payload in a multicast message, and consequently the number of sensors. Besides the protocol itself defines a probing procedure to detect collisions that can impose a waiting time up to 750ms (Sec 8.1 of [42]), which can significantly delay record availability. Figure 5.21b represents the worst case scenario, a single-threaded sequential advertisement of all records, with better times achievable using parallelism and reducing the probing timeouts.

After foreign URIs are advertised, the delay introduced by the FIXP while converting messages was measured. Converting an MQTT subscribe into the NPSN subscribe message introduced a delay of 1,37 ± 0,10ms while converting the NPSN Data message to an MQTT Publish required 1,58 ± 0,17ms.



(a) Discovery and mapping delay

(b) Advertising delay

Figure 5.21: Discovery, mapping and advertising delay

5.4.5 Network overhead analysis

In terms of message size, topic subscription and publication respectively required the exchange of 280 and 271 bytes via MQTT and 621 and 1213 bytes via NPSN. The more substantial amount of exchanged information required in NPSN occurs because NPSN messages contain additional metadata (e.g., signature-related information). Besides, for topic publication, the NPSN followed an Interest-Data-Interest-Data approach to push content leading to a larger number of messages as compared to MQTT.

5.5 Integrated view of discovery topics

In this chapter, different discovery solutions have been presented. These solutions account for the different challenges and scenarios identified in the first section of the chapter. The local domain discovery solution proposed, enables not only discovery in local M2M environments, but also accounts for the local portion of a truly global discovery approach. Using this mechanism, clients may discover brokers which in turn can leverage broker based discovery (e.g., NPSN discovery presented in Section 4.4). Moreover, a mechanism for more flexible discovery through naming semantics was also proposed to account for the unstructured information that characterises the Internet of Things. Finally, to take into consideration the coexistence of not only different protocols but also different networking architectures, an interoperability mechanism is brought into the discovery domain. By integrating these solutions under the specific target scenarios, it is possible to provide a global discovery in the **Named Internet of Things**

CHAPTER 6

Conclusions

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

— Alan Turing

The challenges that the IoT imposes to the ICN networking fabric, altogether with the novelty and early stage of development of this networking paradigm is not a contained topic and the problems and solutions covered in this thesis document span across multiple subtopics. Therefore it is essential to keep these contributions into perspective, to identify open research issues and to outline ideas for future works targeting the new arisen challenges. This final chapter presents the major conclusions drawn from obtained results, along with a global overview on how all pieces fit together into a consistent view on how ICN can be used as an enabler for the IoT. Notwithstanding, the contributions of this PhD thesis offer more than just a set of solutions and results presented in this document. It also opens novel research paths, that could be followed by others, leading to the development of complementary solutions. This fact is evidenced through the high number of citations of outcomes of the presented work. The chapter finalises by highlighting future work to be followed towards a fully-fledged ***"Named Internet of Things"****, as well as to foster the further exploration of the proposed approaches. .*

6.1 Results and Achievements

In this thesis, several aspects related to the development of a **"Named Internet of Things"** were addressed. The primary research efforts towards this end are summarised and analysed (Section 2.3). A comprehensive vision of an ICN-enabled IoT architecture was materialised through an assessment of the benefits and challenges of utilising ICN as a networking solution for the IoT, determining guidelines for its adoption in IoT environments (Section 3.2).

Notably, the potential of ICN contributions in addressing IoT scenarios, compared to IP-based mechanisms is established through simulations, evidencing its benefits and challenges (Section 3.1). Particular emphasis was given to the benefits associated with ICN intrinsic in-network caching mechanism, in terms of network traffic reduction and energy saving. The obtained simulation results demonstrated that by consuming less energy and bandwidth, ICN outperforms IP based scenarios. The trade-off between the available caching storage capacity and the consumption of both energy and bandwidth was established. In this way, it was shown that the deployment of ICN in increasingly prominent areas, such as IoT, can make a positive contribution where IP based designs are otherwise becoming more complex and hard to deploy.

After addressing the suitability of ICN for IoT, and based on the challenges to be faced, the main action points towards the realisation of the **"Named Internet of Things"** were derived. These action points were mainly associated with the management of information freshness and the provision of means for efficient discovery in the envisioned scenarios. Following these guidelines, the core ICN concepts were extended by providing specific solutions that targeted the identified challenges. These enhancements were prototyped and evaluated, demonstrating its practicality and feasibility.

The available NDN mechanism for freshness control was examined in detail, and a new consumer-driven information freshness approach for NDN was proposed to enable consumers to specify the desired freshness for the requested information (Section 4.2). Two simulation scenarios were designed and developed to characterise the NDN mechanism for freshness control and to evaluate the newly proposed mechanism. While ensuring the data quality in environments featuring consumers with different freshness requirements, the new proposed approach showed to be a viable solution to reduce the adverse effects of the freshness on the network performance. Additionally, information freshness management was integrated into a secure management framework enabling the use of freshness as part of a service level agreement (Section 4.3). Continuing into the development of features for a more reliable information freshness, NDN was enhanced with publish-subscribe capabilities in the NPSN protocol (Section 4.4). This protocol serves as a proof-of-concept for showcasing the benefits of using a publish-subscribe approach on top of NDN instead of a polling-based approach in scenarios where information generation time is uncertain (e.g. emergency notification).

With the increased interest for ICN in IoT/M2M environments, the ICN protocol stacks have to account for efficient service discovery. In this context, the NPSN protocol was provided with discovery capabilities. Also, a local domain discovery protocol was designed in such a way that it can be both reactive and proactive, while better leveraging the broadcast nature of wireless media (Section 5.2). In order to support such capabilities, the NDN reference implementation was extended with the notion of an alternative forwarding pipeline. As a result, NDN nodes become able to send and receive Data messages without a matching Interest. The local domain solution could then be used to complement the broker based solution of the NPSN protocol. Finally, the use of semantic similarity mechanisms for a more flexible discovery was proposed, with results showcasing the potential of exploring such solutions (Section 5.3).

All proposed enhancements kept the ICN principles unaltered, ensuring its applicability beyond the scope of the IoT, contributing to increasing the maturity of ICN. Besides, means for ensuring the interoperability of the proposed solutions were proposed to ensure the smooth adoption of these modifications.

Particularly, the integration of the proposed discovery and interoperability mechanisms (Section 5.4) greatly facilitate the communication between endpoints in different network architectures, even if they incur in initial setup costs, paving the way for the smooth adoption of novel networking paradigms and architectures.

6.1.1 Global Conclusions

The emergence of novel utilisation patterns has challenged the current Internet. As for now, an evolutionary approach, based on patching the current IP networking core infrastructure has been followed to cope with these new scenarios. Alternatively, clean slate approaches for the Future Internet have also been explored, with ICN being one of the most prominent proposals.

In particular, the IoT has been proving itself as one of the most challenging technologies of current ICT deployments. At the same time, it has the potential to change the economy and society by transforming aspects ranging from the way of living to business practices. As a consequence, the IoT has captured the spotlight and attracted different actors (e.g., academia, industry, standardisation).

The fact that ICN partially matches the information exchange model of the IoT has triggered research on the potential of this networking infrastructure for supporting current and future IoT scenarios. However, ICN is still far from an IoT-ready networking solution. Notwithstanding, the integration of both technologies could be a positive factor for each of them. While ICN could bring better support for IoT functionalities by incorporating them still at the design stage, IoT could be both the technological trigger for achieving further maturity of a networking paradigm still at its infancy and the economic trigger required to bring down the hegemonic power of IP in the current ICT landscape.

In this context, throughout this document, after a comprehensive overview on the main strengths and shortcomings of the utilisation of ICN as a networking infrastructure for IoT environments, different complementary solutions that build on each other were proposed. As a result, it is possible to visualise a fully fledged framework which has support for both poll-based and publish-subscribe communication models. In either model, the quality of the information is guaranteed while preserving the benefits associated with the intrinsic ICN mechanisms. Aspects such as discovery, security and interoperability are also considered. Particularly the discovery mechanisms account for both local and global scale, ensuring its practicability across the broad spectrum of applications and scenarios expected for IoT environments. Moreover, aspects such as information structure are also considered, allowing on the one hand to break the informational silos originated due to the heterogeneous ways IoT content is shared among different technologies and manufacturers and on the other hand to adequately (dis)aggregate IoT content.

6.2 Future Research Paths

As discussed, the outcomes of the presented work provide the necessary tools for enabling what was dubbed as the **"Named Internet of Things"**. These tools target the challenges of addressing the IoT from an ICN perspective. Notwithstanding, there is room for improvement, and there are several ways the presented work could continue its development. Each of these follow-up approaches leads to a roadmap that enables the further development of each associated technology, as well as the integrated concept. This section presents and discusses some of the most immediate subjects that could follow this research work.

6.2.1 Efficient Retrieval of Named IoT Content

In general, IoT devices provide different information, generating typically small pieces of content. As such, the aggregation of IoT information is a shared vision for IoT content dissemination optimisation. This vision then raises the question of how the ICN functionalities can be better leveraged for the (dis)aggregation of IoT content.

The ICN protocols envision data retrieval where the pieces of Data are requested by name (i.e., a name acts as a data selector of the data distributed over a network). However, consumers may need to express complex queries for data (e.g. the temperature perceived by all sensors in a given room). Such a feature requires that the producers can accept and handle such queries, based on standardised or custom query formats, and return the desired subset of the data.

A key benefit commonly associated with ICN is the intrinsic availability of network caching, but granular caching is only available if the content producer publishes data with the desired granularity. As such, while this type of dis(aggregation) could largely fit IoT concepts, it attempts against the benefits of the ICN caching mechanism.

Moreover, a desirable consequence of distributing the content over the network is to offload work over this data across the network. To manipulate data, one often uses some form of data manipulation language that matches the underlying data model (e.g., record databases support some version of SQL, graph structures have GRAPHQL, and other data structures have similar structured query languages).

In this context, two main research issues may be established: (i) how to provide a way of selectively accessing specific parts of the content from within a given content, enabling the (dis)aggregation for IoT content while still leveraging ICN in-network caching?; and (ii) how to include the network itself in the (dis)aggregation process?

To exemplify this idea, lets consider a scenario, like the one shown in Figure 6.1, where the information from different weather sensors (i.e., X, Y and Z) is aggregated into a *"Weather"* content. However, different users may be interested in different parts of that content (e.g., Weather, Wind, Wind from Sensor Z).

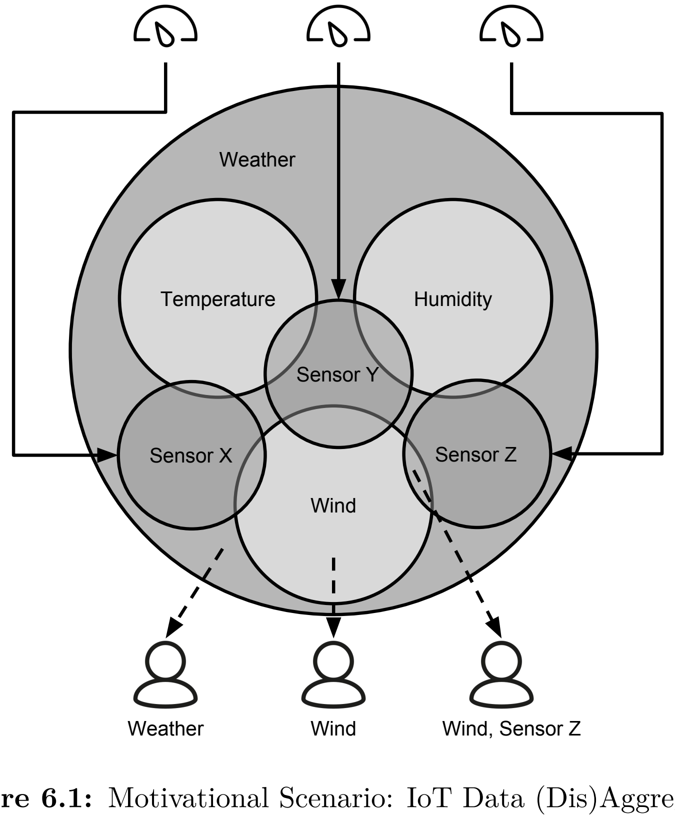


Figure 6.1: Motivational Scenario: IoT Data (Dis)Aggregation

In NDN, as shown in Figure 6.2, an individual request for a disaggregated piece of content has to go straight to the source of information even though the information is already available in the aggregated content stored in the intermediary routers (i.e., in-network caching), since there is no direct match with the requested information.

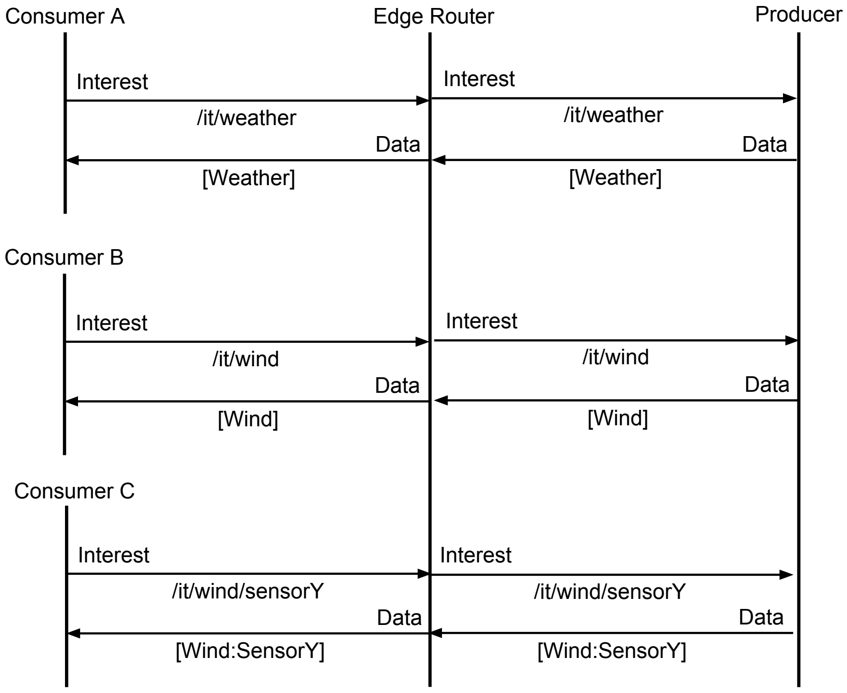


Figure 6.2: Signalling: Simple Request

By adding intelligence to intermediary routers, it is possible to avoid requests to unnecessarily be forwarded up to the source by extracting the requested information from the aggregated data available at the content stores as shown in Figure 6.3.

However, as shown in Figure 6.4, the order in which requests are received at the intermediary routers will also impact the retrieval mechanism. If contents are requested from the more disaggregated to the more aggregated, requests will still have to go up to the source to retrieve the content.

To avoid this issue, a different approach, in which both the intermediary routers and the producers agree to participate in an advanced response, is required. The goal is to allow the proactive push of larger pieces of information for better leveraging disaggregation mechanisms, as shown in Figure 6.5. While the first request received at the Edge Router and forwarded to the Producer is for `/it/wind/sensorY`, the Producer replies with a more complete piece of information (`/it/weather`). This response is cached and further disaggregated at the Edge Router, which in turn replies to the Consumer (Consumer A) with the initially requested piece of information.

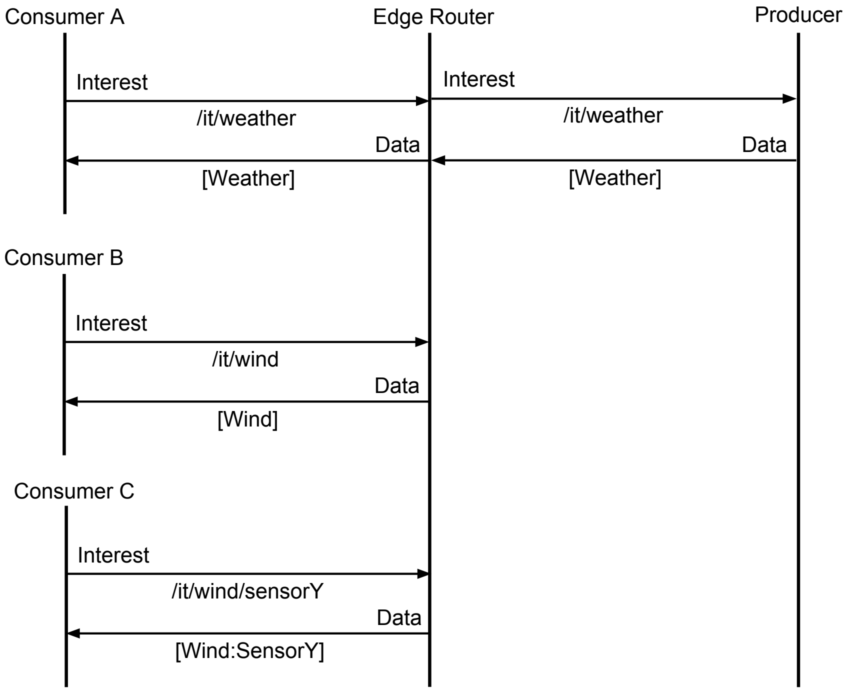


Figure 6.3: Signalling: Advanced Request

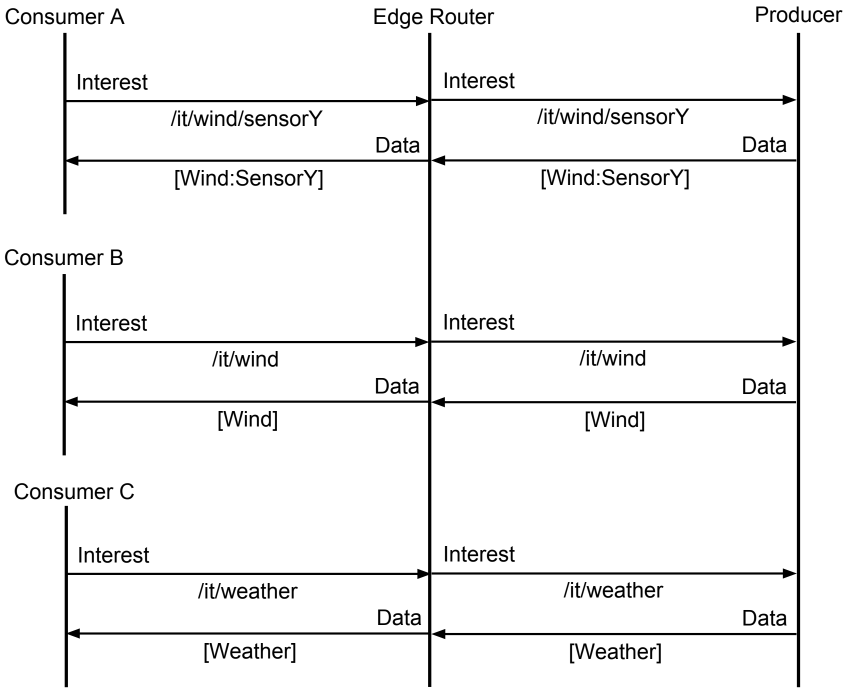


Figure 6.4: Signalling: Simple Response

In realising such a mechanism, different strategies could be followed. These strategies, as summarised in Figure 6.6, can be categorised as follows:

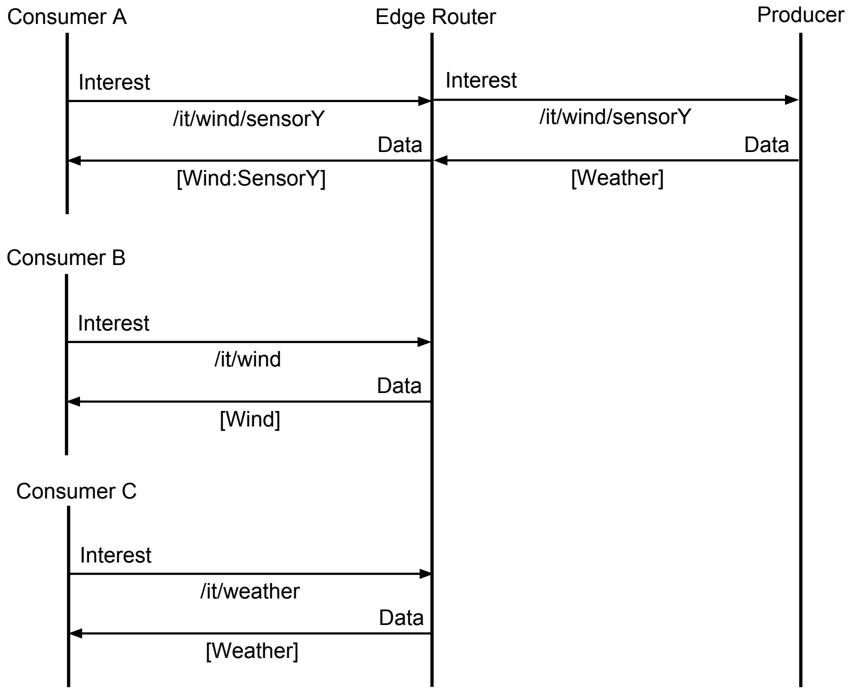


Figure 6.5: Signalling: Advanced Response

- Source of the (dis)aggregation information:
  - The producer provides detailed content (dis)aggregation information (e.g., manifest);
  - A network entity has the content (dis)aggregation intelligence (e.g., parsers).
- Holder of the (dis)aggregation information:
  - Consumer: Consumers hold the (dis)aggregation information and generate complex queries which identify the desired piece of content. For such an approach to work the intermediary routers must implement and support a (dis)aggregation protocol (e.g. byte selection based on Interest parameters);
  - Router: Intermediary routers hold the (dis)aggregation information and process incoming Interests returning the associated data. Users should have the required knowledge to request pieces of content (e.g., through discovery procedures);
  - Third Party: A third party entity holds the (dis)aggregation information. In this approach, the (dis)aggregation duties are offloaded to the third party entity which must implement a (dis)aggregation protocol known to users and routers.

Hybrid approaches in which both consumers and routers hold the (dis)aggregation information are also envisioned.

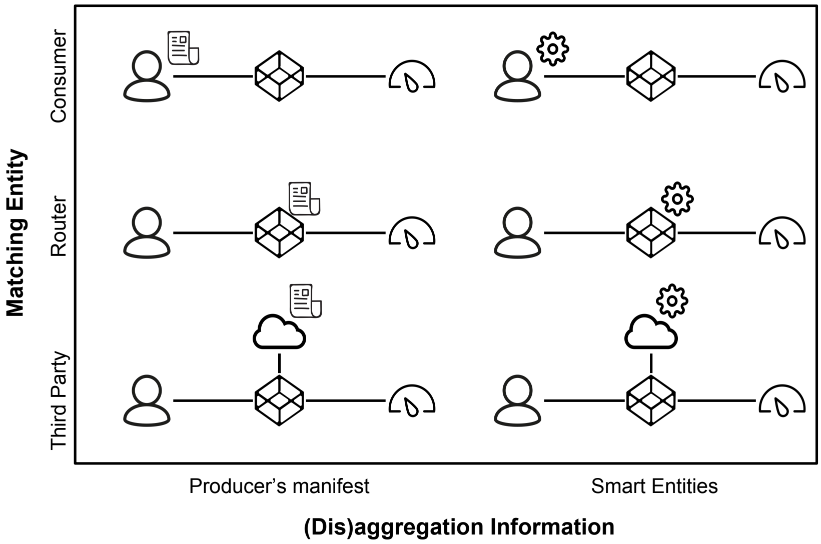


Figure 6.6: Disaggregation strategies

A preliminary implementation of this concept was developed in [30] as part of the SeLF-ICN project, featuring a security surveillance scenario, where the ICN networking layer was enhanced to not only reach for content using names but actually to specify and retrieve particular elements within the content itself. In doing so, ICN and Light Field (LF) Imaging technologies were combined for enabling the selective retrieval of specific images based on metadata information about the features contained in the images.

6.2.2 Cloudification for an Efficient Named Interest of Things

This section motivates the principles for the utilisation of ICN and virtualisation, to enable the adequate operation of large scale IoT deployments. The goal is to allow an IoT service provider to support the scalable access of multiple clients to its physical IoT infrastructure. Additionally, the proposal accounts for the integration of heterogeneous devices with different resource constraints, supporting diverse communication technologies and protocols. Moreover, it enables an entirely distributed approach, where management functionalities (e.g., service discovery, access control) are decoupled from the execution of operations over the physical infrastructure. Besides, the proposed architecture can accommodate an increasing number of IoT devices, as each of them is managed using a virtual representation of the device that is instantiated as a software component in the cloud.

6.2.2.1 Main concepts and operational principles

An overview of a framework realising the Cloud-based Named Internet of Things is presented in Figure 6.7. The different elements of the framework, along with the associated and functionalities, can be summarised as follows:

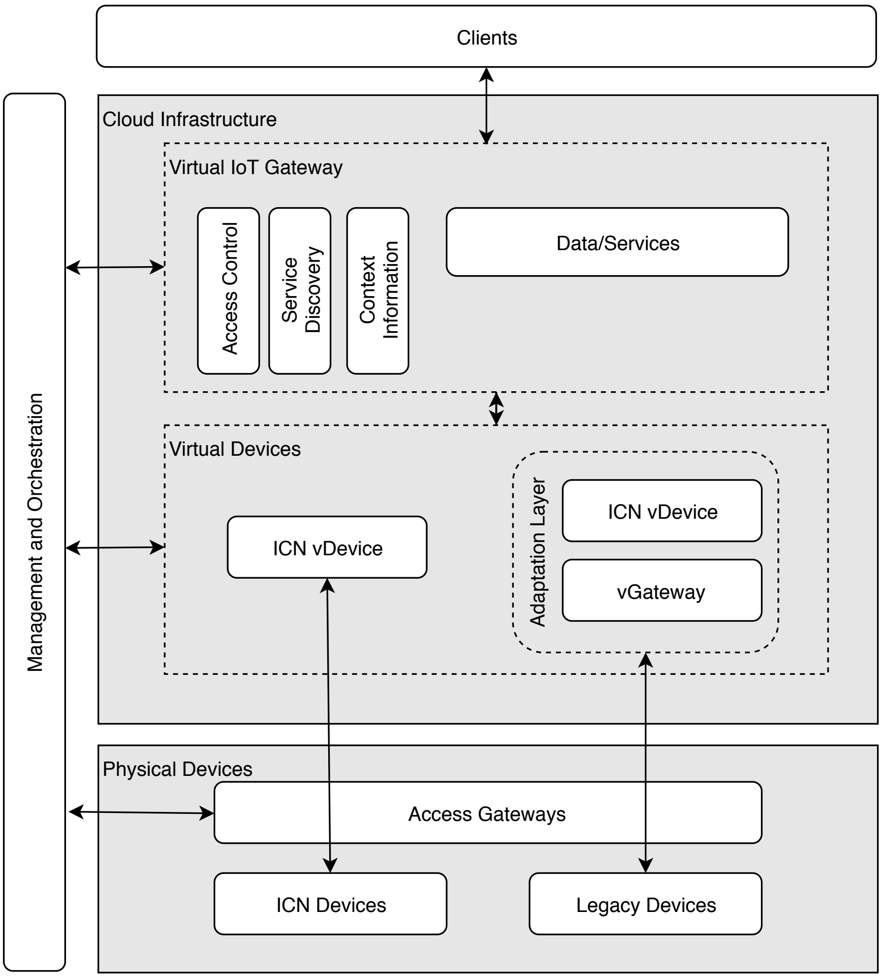


Figure 6.7: Cloud-based Named Internet of Things

Physical Devices

The physical devices include, as its name indicates, all devices taking part in the solution: (i) IoT devices and (ii) Access Gateways.

IoT devices represent any physical device providing relevant data, either sensing information or the result of the execution of an actuation command. These devices form a potentially large and heterogeneous network of entities with disparate characteristics (e.g., constraints in terms of memory, processor, energy consumption). Moreover, these devices may support, as indicated, different communication technologies and protocols (e.g., ICN, MQTT, CoAP). Devices not supporting ICN will be referenced hereafter as Legacy Devices.

Access Gateways are the devices providing network access for the IoT devices to communicate with the cloud. Additionally, these gateways interact with the Management and Orchestration (MANO) to notify the status of the connected IoT devices, consequently triggering the required actions (e.g., virtual device instantiation).

Cloud Infrastructure

The cloud infrastructure represents all the virtualised entities which fall into two main concepts: (i) Virtual Devices and (ii) Virtual IoT Gateway.

Virtual devices are software representations of their physical counterpart. MANO instantiates every physical device connected through an Access Gateway as virtual devices, which ultimately provides the sensing information or receive the control information via ICN protocols. As such, they act as an intermediary entity in the data exchange between consumers and actual devices. Consequently, the virtual device is exposed as the point of contact to access the physical device, and any entity accessing the solution sees the virtual device as the physical device itself. Virtual devices may also implement any further mechanism to be provided by the resulting IoT framework. In doing so, physical devices may be relieved, in accordance to their constraints, form the burden of keeping per-client information, executing per-client procedures (e.g., encryption, authentication and client authorisation), this way enhancing scalability. The communication between the virtual and the physical device can be protected through encryption and authentication, depending on the capacities supported by the physical device. Finally, legacy devices, as previously mentioned, may implement different communication technologies and protocols, not necessarily following an information-centric approach (e.g., CoAP or MQTT). In these cases, an adaptation process is required, and the virtual devices act as an entity enabling the interaction of clients with these heterogeneous devices, by supporting the interoperability mechanisms presented in this work and detailed in [25].

Virtual IoT gateways represent the actual point of interaction of the clients. In doing so, it is possible to increase even further the scalability of the solution by providing a way of grouping several virtual devices and providing not only common functionalities, such as service discovery and access control but also enabling data processing and aggregation as the exposure of compound information as an IoT service. IoT services provide advanced functionalities by interacting with one or multiple virtual devices. These services could be accessed using the same communication mechanisms as for raw data information of virtual devices by being exposed to clients as conventional IoT devices made accessible by the IoT gateway. Deploying these virtual gateways at the Mobile Edge Computing (MEC) could be a plausible approach in cases where low latency is required.

Management and Orchestration

MANO provides the support for discovery and registration of physical devices connected through the access gateways. After the successful registration of an IoT device, MANO instantiates a virtual device with all the associated requirements (e.g., interoperability mechanisms). Additionally, MANO is responsible for instantiating new IoT virtual gateways as required as well as the services provided by specific gateways. MANO control and configuration should be the responsibility of the IoT provider.

6.3 Final Thoughts

As a whole, this thesis is expected to contribute to the understanding of the trends, benefits, and challenges of the next generation of networks, creating precedents for further innovations, while proposes solutions to the identified shortcomings. The current section sheds some final thoughts on the realisation of the vision advocated through the thesis and contextualises, them in the advent of the fifth generation network systems.

ICN is a novel networking paradigm still at its early stages and lacks the maturity that the years of optimisations and the extensive involvement of the research community have shed on the current Internet networking solutions. Although IP has followed a patching approach, and its ability to cope with today's and future networking usage patterns is highly questionable, it is deeply rooted in the current Internet. This challenges novel networking solutions that may arise. Consequently, enabling a smooth roll-out of ICN solutions will undoubtedly be a fundamental aspect of its adoption.

It is envisioned that ICN interoperability will increasingly become a key research focus, mainly because the Internet is continuously evolving, and a new generation of networks is arising encompassing new communication scenarios and verticals, in a flexible, dynamic and cost-effective way. It could be argued that ICN could become part of 5G-based solutions and that it will keep pushing original new utilisation use cases in Beyond-5G research.

5G networks are expected to account for different use cases, each with tailored requirements. For example, Massive Machine-Type Communications (mMTC), is identified as one of the 5G main usage scenarios, which involves the support of high-density, highly heterogeneous devices and services with very stringent requirements (e.g., availability, reliability, latency, energy-efficiency). In this regard, network slicing appears as a crucial enabler of the 5G concept, allowing multiple logical networks to run on a shared infrastructure. The idea is to enable the independent operation of different use cases with varying requirements through the use of tailored network slices. SDN and Network Function Virtualization (NFV) technologies are considered to be essential in the realisation of the slicing concept.

In this context, network slicing could ensure a smooth introduction of ICN into the core of the 5G networks in a way that is economically and operationally feasible, acting as a double enabler. First, by creating the means for a native deployment of ICN, it contributes to the continuous development of this novel networking paradigm. Second, the core advantages of the ICN concept can be brought to life to overcome the limitations of other protocols and satisfy the necessities of the different use cases expected to be part of the 5G networks. Notably, mMTC use cases could leverage the concepts developed in this thesis and by the research community, in general, through the use of ICN dedicated slices. However, with 5G, and with ICN as well, many new other areas and verticals could be explored towards the delivery of the Future Internet.

References

[1] Cisco. *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*. Tech. rep. Cisco, 2018.

[2] S. Shenker. "Fundamental design issues for the future Internet". In: *IEEE Journal on Selected Areas in Communications* 13.7 (Sept. 1995), pp. 1176–1188. ISSN: 0733-8716. DOI: 10.1109/49.414637.

[3] M. Handley. "Why the Internet only just works". In: *BT Technology Journal* 24.3 (July 2006), pp. 119–129. ISSN: 1573-1995. DOI: 10.1007/s10550-006-0084-z. URL: https://doi.org/10.1007/s10550-006-0084-z.

[4] J. Pan, S. Paul, and R. Jain. "A survey of the research on future internet architectures". In: *IEEE Communications Magazine* 49.7 (July 2011), pp. 26–36. ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5936152.

[5] Darleen Fisher. "A Look Behind the Future Internet Architectures Efforts". In: *SIGCOMM Comput. Commun. Rev.* 44.3 (July 2014), pp. 45–49. ISSN: 0146-4833. DOI: 10.1145/2656877.2656884. URL: http://doi.acm.org/10.1145/2656877.2656884.

[6] Jennifer Rexford and Constantine Dovrolis. "Future Internet architecture: clean-slate versus evolutionary research". In: *Communications of the ACM* 53.9 (2010), pp. 36–40.

[7] Anja Feldmann. "Internet Clean-slate Design: What and Why?" In: *SIGCOMM Comput. Commun. Rev.* 37.3 (July 2007), pp. 59–64. ISSN: 0146-4833. DOI: 10.1145/1273445.1273453. URL: http://doi.acm.org/10.1145/1273445.1273453.

[8] Michio Honda, Yoshifumi Nishida, Costin Raiciu, Adam Greenhalgh, Mark Handley, and Hideyuki Tokuda. "Is It Still Possible to Extend TCP?" In: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. IMC '11. Berlin, Germany: ACM, 2011, pp. 181–194. ISBN: 978-1-4503-1013-0. DOI: 10.1145/2068816.2068834. URL: http://doi.acm.org/10.1145/2068816.2068834.

[9] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. "A survey of information-centric networking". In: *IEEE Communications Magazine* 50.7 (July 2012), pp. 26–36. ISSN: 0163-6804. DOI: 10.1109/MCOM.2012.6231276.

[10] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. "A Survey of Information-Centric Networking Research". In: *IEEE Communications Surveys Tutorials* 16.2 (Second 2014), pp. 1024–1049. ISSN: 1553-877X. DOI: 10.1109/SURV.2013.070813.00063.

[11] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies, A. Molinaro, and S. Eum. *Information-Centric Networking: Baseline Scenarios*. RFC 7476. IETF, Mar. 2015. URL: http://tools.ietf.org/rfc/rfc7476.txt.

[12] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (Oct. 2010), pp. 2787–2805. ISSN: 1389-1286. DOI: 10.1016/J.COMNET.2010.05.010. URL: https://www.sciencedirect.com/science/article/pii/S1389128610001568.

[13] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. "Internet of things: Vision, applications and research challenges". In: *Ad Hoc Networks* 10.7 (Sept. 2012), pp. 1497–1516. ISSN: 1570-8705. DOI: 10.1016/J.ADHOC.2012.02.016. URL: https://www.sciencedirect.com/science/article/pii/S1570870512000674.

[14] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future Generation Computer Systems* 29.7 (Sept. 2013), pp. 1645–1660. ISSN: 0167-739X. DOI: 10.1016/J.FUTURE.2013.01.010. URL: https://www.sciencedirect.com/science/article/pii/S0167739X13000241.

[15] J. A. Stankovic. "Research Directions for the Internet of Things". In: *IEEE Internet of Things Journal* 1.1 (Feb. 2014), pp. 3–9. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2312291.

[16] D. Singh, G. Tripathi, and A. J. Jara. "A survey of Internet-of-Things: Future vision, architecture, challenges and services". In: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. Mar. 2014, pp. 287–292. DOI: 10.1109/WF-IoT.2014.6803174.

[17] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges". In: *2012 10th International Conference on Frontiers of Information Technology*. Dec. 2012, pp. 257–260. DOI: 10.1109/FIT.2012.53.

[18] Wentao Shang, Yingdi Yu, Ralph Droms, and Lixia Zhang. "Challenges in IoT networking via TCP/IP architecture". In: *NDN, Technical Report NDN-0038* (2016).

[19] Ravi Ravindran, Yanyong Zhang, Luigi Alfredo Grieco, Anders Lindgren, Jeff Burke, Bengt Ahlgren, and Aytac Azgin. *Design Considerations for Applying ICN to IoT*. Internet-Draft draft-irtf-icnrg-icniot-02. Work in Progress. Internet Engineering Task Force, Oct. 2018. 51 pp. URL: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-icniot-02.

[20] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos. "Information-centric networking for the internet of things: challenges and opportunities". In: *IEEE Network* 30.2 (Mar. 2016), pp. 92–100. ISSN: 0890-8044.

[21] J. Quevedo, D. Corujo, and R. L. Aguiar. "A case for ICN usage in IoT environments". In: *2014 IEEE Global Communications Conference*. Dec. 2014, pp. 2770–2775.

[22] J. Quevedo, D. Corujo, and R. L. Aguiar. "Consumer driven information freshness approach for content centric networking". In: *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Apr. 2014, pp. 482–487.

[23] J. Suarez, J. Quevedo, I. Vidal, D. Corujo, J. Garcia-Reinoso, and R. L. Aguiar. "A secure IoT management architecture based on Information-Centric Networking". In: *Journal of Network and Computer Applications* 63 (2016), pp. 190–204. ISSN: 1084-8045.

[24] D. Corujo, C. Guimarães, J. Quevedo, R. Ferreira, and R. L. Aguiar. "Information Centric Exchange Mechanisms for IoT Interoperable Deployment". In: *User-Centric and Information-Centric Networking and Services: Access Networks and Emerging Trends*. Ed. by M.B. Krishna. Under Submission. Taylor & Francis Group, 2018.

[25] Carlos Guimarães, José Quevedo, Rui Ferreira, Daniel Corujo, and Rui L. Aguiar. "Exploring interoperability assessment for Future Internet Architectures roll out". In: *Journal of Network and Computer Applications* (2019). ISSN: 1084-8045.

[26] J. Quevedo, R. Ferreira, C. Guimarães, R. L. Aguiar, and D. Corujo. "Internet of Things discovery in interoperable Information Centric and IP networks". In: *Internet Technology Letters* 1.1 (). e1 ITL-17-0001.R1, e1.

[27] Carlos Guimarães, José Quevedo, Rui Ferreira, Daniel Corujo, and Rui L. Aguiar. "Content Retrieval While Moving Across IP and NDN Network Architectures". In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. June 2019.

[28] J. Quevedo, M. Antunes, D. Corujo, D. Gomes, and R. L. Aguiar. “On the application of contextual IoT service discovery in Information Centric Networks”. In: *Computer Communications* 89-90 (2016). Internet of Things Research challenges and Solutions, pp. 117–127. ISSN: 0140-3664.

[29] J. Quevedo, C. Guimarães, R. Ferreira, D. Corujo, and R. L. Aguiar. “ICN as Network Infrastructure for Multi-Sensory Devices: Local Domain Service Discovery for ICN-based IoT Environments”. In: *Wireless Personal Communications* 95.1 (July 2017), pp. 7–26. ISSN: 1572-834X.

[30] J. Quevedo, C. Guimarães, R. Ferreira, A. Sepas-Moghaddam, L. Malhadas, R. L. Aguiar, P. L. Correia, and D. Corujo. “Selectively Accessing Light Field Face Images over Information Centric Networking”. In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. Feb. 2018, pp. 1–5.

[31] Kevin Ashton. “That ’Internet of Things’ Thing”. In: *RFiD Journal* 22 (2009), pp. 97–114.

[32] Yousaf Bin Zikria, Muhammad Khalil Afzal, Farruh Ishmanov, Sung Won Kim, and Heejung Yu. “A survey on routing protocols supported by the Contiki Internet of things operating system”. In: *Future Generation Computer Systems* 82 (2018), pp. 200–219. ISSN: 0167-739X. DOI: `https://doi.org/10.1016/j.future.2017.12.045`. URL: `http://www.sciencedirect.com/science/article/pii/S0167739X17324299`.

[33] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. In: *IEEE Communications Surveys Tutorials* 17.4 (Fourthquarter 2015), pp. 2347–2376. ISSN: 1553-877X. DOI: `10.1109/COMST.2015.2444095`.

[34] Ammar Rayes and Samer Salam. “IoT Protocol Stack: A Layered View”. In: *Internet of Things From Hype to Reality: The Road to Digitization*. Cham: Springer International Publishing, 2019, pp. 103–154. ISBN: 978-3-319-99516-8. DOI: `10.1007/978-3-319-99516-8_5`. URL: `https://doi.org/10.1007/978-3-319-99516-8_5`.

[35] “IEEE Standard for Low-Rate Wireless Networks”. In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (Apr. 2016), pp. 1–709. DOI: `10.1109/IEEESTD.2016.7460875`.

[36] N. Kushalnagar, G. Montenegro, and C. Schumacher. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*. RFC 4919. IETF, Aug. 2007. URL: `http://tools.ietf.org/rfc/rfc4919.txt`.

[37] S. Farrell. *Low-Power Wide Area Network (LPWAN) Overview*. RFC 8376. IETF, May 2018. URL: `http://tools.ietf.org/rfc/rfc8376.txt`.

[38] Y. D. Beyene, R. Jantti, O. Tirkkonen, K. Ruttik, S. Iraji, A. Larmo, T. Tirronen, and a. J. Torsner. “NB-IoT Technology Overview and Experience from Cloud-RAN Implementation”. In: *IEEE Wireless Communications* 24.3 (June 2017), pp. 26–32. ISSN: 1536-1284. DOI: `10.1109/MWC.2017.1600418`.

[39] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. RFC 4944. IETF, Sept. 2007. URL: `http://tools.ietf.org/rfc/rfc4944.txt`.

[40] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. IETF, Mar. 2012. URL: `http://tools.ietf.org/rfc/rfc6550.txt`.

[41] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann. *Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)*. RFC 6775. IETF, Nov. 2012. URL: `http://tools.ietf.org/rfc/rfc6775.txt`.

[42] S. Cheshire and M. Krochmal. *Multicast DNS.* RFC 6762. IETF, Feb. 2013. URL: `http://tools.ietf.org/rfc/rfc6762.txt`.

[43] E. Rescorla and N. Modadugu. *Datagram Transport Layer Security Version 1.2.* RFC 6347. IETF, Jan. 2012. URL: `http://tools.ietf.org/rfc/rfc6347.txt`.

[44] Andrew Banks and Rahul Gupta. *MQTT Version 3.1.1.* Tech. rep. Apr. 2015.

[45] OASIS. *Advanced Message Queuing Protocol (AMQP) Version 1.0.* Tech. rep. Oct. 2012.

[46] Z. Shelby, K. Hartke, and C. Bormann. *The Constrained Application Protocol (CoAP).* RFC 7252. IETF, June 2014. URL: `http://tools.ietf.org/rfc/rfc7252.txt`.

[47] N. Naik. "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP". In: *2017 IEEE International Systems Engineering Symposium (ISSE).* Oct. 2017, pp. 1–7. DOI: `10.1109/SysEng.2017.8088251`.

[48] Roy T Fielding. *Architectural styles and the design of network-based software architectures.* Vol. 7. University of California, Irvine Doctoral dissertation, 2000.

[49] Rui L. Aguiar. "Some comments on hourglasses". In: *SIGCOMM Comput. Commun. Rev.* 38.5 (Sept. 2008), pp. 69–72. ISSN: 0146-4833. DOI: `10.1145/1452335.1452346`.

[50] C. Perkins. *IP Mobility Support.* RFC 2002. IETF, Oct. 1996. URL: `http://tools.ietf.org/rfc/rfc2002.txt`.

[51] Y. Zou and K. Chakrabarty. "Distributed Mobility Management for Target Tracking in Mobile Sensor Networks". In: *IEEE Transactions on Mobile Computing* 6.8 (Aug. 2007), pp. 872–887. ISSN: 1536-1233. DOI: `10.1109/TMC.2007.1005`.

[52] David R Cheriton and Mark Gritter. "TRIAD: A new next-generation Internet architecture". In: (2000).

[53] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. "A data-oriented (and beyond) network architecture". In: *SIGCOMM Comput. Commun. Rev.* 37.4 (Aug. 2007), pp. 181–192. ISSN: 0146-4833. DOI: `10.1145/1282427.1282402`.

[54] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. "Networking named content". In: *Proceedings of the 5th international conference on Emerging networking experiments and technologies.* CoNEXT '09. Rome, Italy: ACM, 2009, pp. 1–12. ISBN: 978-1-60558-636-6. DOI: `10.1145/1658939.1658941`.

[55] Dirk Trossen, George Parisis, Borislava Gajic, Janne Riihijarvi, Paris Flegkas, Pasi Sarolahti, Petri Jokela, Xenofon Vasilakos, Christos Tsilopoulos, Somaya Arianfar, and Martin Reed. *Architecture Definition, Components Descriptions and Requirements.* Tech. rep. PURSUIT, 2011.

[56] Christian Dannewitz, Dirk Kutscher, Börje Ohlman, Stephen Farrell, Bengt Ahlgren, and Holger Karl. "Network of Information (NetInf) - An information-centric networking architecture". In: *Computer Communications* 0 (2013), pp. 1–2. ISSN: 0140-3664. DOI: `10.1016/j.comcom.2013.01.009`.

[57] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. "Named Data Networking". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73.

[58] Bastiaan Wissingh, Christopher A. Wood, Alex Afanasyev, Lixia Zhang, David R. Oran, and Christian Tschudin. *Information-Centric Networking (ICN): CCN and NDN Terminology.* Internet-Draft draft-irtf-icnrg-terminology-03. Work in Progress. Internet Engineering Task

Force, Mar. 2019. 18 pp. URL: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-terminology-03.

[59] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moi-seenko, Yingdi Yu, Wentao Shang, Yi Huang, Jerald Paul Abraham, Steve DiBenedetto, et al. *NFD developers guide.* Tech. rep. Technical Report NDN-0021, NDN Project, 2014.

[60] John Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. "Building efficient wireless sensor networks with low-level naming". In: *ACM SIGOPS Operating Systems Review.* Vol. 35. 5. ACM. 2001, pp. 146–159.

[61] Daniel Corujo, Rui L. Aguiar, Iván Vidal, Jaime García-Reinoso, and Kostas 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.

[62] Yuning Song, Huadong Ma, and Liang Liu. "Content-Centric Internetworking for Resource-Constrained Devices in the Internet of Things". In: (2013).

[63] Jeff Burke. "Authoring Place-based Experiences with an Internet of Things: Tussles of Expressive, Operational, and Participatory Goals". In: (2011).

[64] J Burke, A Horn, and A Marianantoni. *Authenticated lighting control using named data networking.* Tech. rep. Technical report, UCLA, 2012.

[65] Trisha Biswas, Asit Chakraborti, Ravishankar Ravindran, Xinwen Zhang, and Guoqiang Wang. "Contextualized Information-Centric Home Network". In: (2013).

[66] R. Ravindran, T. Biswas, Xinwen Zhang, A. Chakraborti, and Guoqiang Wang. "Information-centric networking based homenet". In: *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on.* May 2013, pp. 1102–1108.

[67] Ngoc-Thanh Dinh and Younghan 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.

[68] Bilel Saadallah, Abdelkader Lahmadi, Olivier Festor, et al. "CCNx for Contiki: implementation details". In: (2012).

[69] Zhong Ren, M.A. Hail, and 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.

[70] Torsten Teubler, MohamedAhmedM. Hail, and Horst Hellbrück. "Efficient Data Aggregation with CCNx in Wireless Sensor Networks". In: *Advances in Communication Networking.* Ed. by Thomas Bauschert. Vol. 8115. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 209–220. ISBN: 978-3-642-40551-8. DOI: 10.1007/978-3-642-40552-5_19.

[71] Lucas Wang, Alexander Afanasyev, Romain Kuntz, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. "Rapid traffic information dissemination using named data". In: *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications.* ACM. 2012, pp. 7–12.

[72] Lucas Wang, Ryuji Wakikawa, Romain Kuntz, Rama Vuyyuru, and Lixia Zhang. "Data naming in vehicle-to-vehicle communications". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on.* IEEE. 2012, pp. 328–333.

[73] D. Corujo, Rui L. Aguiar, I. Vidal, and J. Garcia-Reinoso. "A named data networking flexible framework for management communications". In: *IEEE Communications Magazine* 50.12 (2012), pp. 36–43. ISSN: 0163-6804. DOI: 10.1109/MCOM.2012.6384449.

[74] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. “Multi-source Data Retrieval in IoT via Named Data Networking”. In: *Proceedings of the 1st International Conference on Information-centric Networking*. INC ’14. Paris, France: ACM, 2014, pp. 67–76. ISBN: 978-1-4503-3206-4.

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

[76] Jeff Burke et al. “Secure Sensing over Named Data Networking”. In: *IEEE Network Computing and Applications (NCA)*. 2014, pp. 175–180.

[77] Emmanuel Baccelli, Christian Mehlis, Oliver Hahm, Thomas C. Schmidt, and Matthias 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. ISBN: 9781450332064. DOI: 10.1145/2660129.2660144. arXiv: 1406.6608.

[78] Cenk Gündoğan, Thomas C. Schmidt, Matthias Wählisch, Christopher Scherb, Claudio Marxer, and Christian Tschudin. *ICN Adaptation to LowPAN Networks (ICN LoWPAN)*. Internet-Draft draft-irtf-icnrg-icnlowpan-02. Work in Progress. Internet Engineering Task Force, Mar. 2019. 42 pp. URL: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-icnlowpan-02.

[79] Guoqiang Zhang, Yang Li, and Tao Lin. “Caching in information centric networking: A survey”. In: *Computer Networks* 57.16 (2013). Information Centric Networking, pp. 3128–3141. ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/j.comnet.2013.07.007.

[80] Luca Muscariello, Giovanna Carofiglio, and Massimo Gallo. “Bandwidth and Storage Sharing Performance in Information Centric Networking”. In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*. ICN ’11. Toronto, Ontario, Canada: ACM, 2011, pp. 26–31. ISBN: 978-1-4503-0801-4. DOI: 10.1145/2018584.2018593.

[81] Uichin Lee, I. Rimac, D. Kilper, and V. Hilt. “Toward energy-efficient content dissemination”. In: *IEEE Network* 25.2 (Mar. 2011), pp. 14–19. ISSN: 0890-8044. DOI: 10.1109/MNET.2011.5730523.

[82] Jun Li, Bin Liu, and Hao Wu. “Energy-Efficient In-Network Caching for Content-Centric Networking”. In: *IEEE Communications Letters* 17.4 (Apr. 2013), pp. 797–800. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2013.022213.122741.

[83] Nakjung Choi, K. Guan, D.C. Kilper, and G. Atkinson. “In-network caching effect on optimal energy consumption in content-centric networking”. In: *Communications (ICC), 2012 IEEE International Conference on*. June 2012, pp. 2889–2894. DOI: 10.1109/ICC.2012.6364320.

[84] J. Llorca, A.M. Tulino, K. Guan, J. Esteban, M. Varvello, Nakjung Choi, and D.C. Kilper. “Dynamic in-network caching for energy efficient content delivery”. In: *INFOCOM, 2013 Proceedings IEEE*. Apr. 2013, pp. 245–249. DOI: 10.1109/INFCOM.2013.6566772.

[85] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. *ndnSIM: NDN simulator for NS-3*. Tech. rep. NDN-0005. NDN, Oct. 2012.

[86] He Wu, Sidharth Nabar, and Radha Poovendran. “An Energy Framework for the Network Simulator 3 (NS-3)”. In: *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. SIMUTools ’11. Barcelona, Spain: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 222–230. ISBN: 978-1-936968-00-8.

[87] M. Stoffers and G. Riley. “Comparing the ns-3 Propagation Models”. In: *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Aug. 2012, pp. 61–67. DOI: 10.1109/MASCOTS.2012.17.

[88] M. Antunes, D. Gomes, and R. Aguiar. “Semantic features for context organization”. In: *Future Internet of Things and Cloud (FiCloud), 2015 International Conference on.* Aug. 2015.

[89] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi, and A. Detti. “Supporting information-centric functionality in software defined networks”. In: *2012 IEEE International Conference on Communications (ICC).* June 2012, pp. 6645–6650. DOI: 10.1109/ICC.2012.6364916.

[90] Salah-Eddine Elayoubi and James Roberts. “Performance and Cost Effectiveness of Caching in Mobile Access Networks”. In: *Proceedings of the 2Nd ACM Conference on Information-Centric Networking.* ACM-ICN ’15. San Francisco, California, USA: ACM, 2015, pp. 79–88. ISBN: 978-1-4503-3855-4. DOI: 10.1145/2810156.2810168. URL: http://doi.acm.org/10.1145/2810156.2810168.

[91] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine. “A Survey of Security Attacks in Information-Centric Networking”. In: *IEEE Communications Surveys Tutorials* 17.3 (thirdquarter 2015), pp. 1441–1454. ISSN: 1553-877X. DOI: 10.1109/COMST.2015.2392629.

[92] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang. “An Overview of Security Support in Named Data Networking”. In: *IEEE Communications Magazine* 56.11 (Nov. 2018), pp. 62–68. ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1701147.

[93] Alexander Afanasyev, J. Alex Halderman, Scott Ruoti, Kent Seamons, Yingdi Yu, Daniel Zappala, and Lixia Zhang. “Content-based Security for the Web”. In: *Proceedings of the 2016 New Security Paradigms Workshop.* NSPW ’16. Granby, Colorado, USA: ACM, 2016, pp. 49–60. ISBN: 978-1-4503-4813-3. DOI: 10.1145/3011883.3011890. URL: http://doi.acm.org/10.1145/3011883.3011890.

[94] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. “Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions”. In: *Journal of Ambient Intelligence and Humanized Computing* (May 2017). ISSN: 1868-5145. DOI: 10.1007/s12652-017-0494-4. URL: https://doi.org/10.1007/s12652-017-0494-4.

[95] Zhiyi Zhang, Yingdi Yu, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. “NAC: Name-based Access Control in Named Data Networking”. In: *Proceedings of the 4th ACM Conference on Information-Centric Networking.* ICN ’17. Berlin, Germany: ACM, 2017, pp. 186–187. ISBN: 978-1-4503-5122-5. DOI: 10.1145/3125719.3132102. URL: http://doi.acm.org/10.1145/3125719.3132102.

[96] Akhila Rao, Olov Schelén, and Anders Lindgren. “Performance Implications for IoT over Information Centric Networks”. In: *Proceedings of the Eleventh ACM Workshop on Challenged Networks.* CHANTS ’16. New York City, New York: ACM, 2016, pp. 57–62. ISBN: 978-1-4503-4256-8. DOI: 10.1145/2979683.2979686. URL: http://doi.acm.org/10.1145/2979683.2979686.

[97] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang. “Named Data Networking of Things (Invited Paper)”. In: *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI).* Apr. 2016, pp. 117–128. DOI: 10.1109/IoTDI.2015.44.

[98] Marica Amadeo, Claudia Campolo, Antonio Iera, and Antonella Molinaro. “Named Data Networking for IoT: an Architectural Perspective”. In: *European Conference on Networks and Communications (EuCNC).* Bologna, Italy, 2014.

[99] Hao Wu, Junxiao Shi, Yaxuan Wang, Yilun Wang, Gong Zhang, Yi Wang, Bin Liu, and Beichuan Zhang. “On Incremental Deployment of Named Data Networking in Local Area Networks”. In: *Proceedings of the Symposium on Architectures for Networking and Communications Systems.* ANCS ’17. Beijing, China: IEEE Press, 2017, pp. 82–94. ISBN: 978-1-5090-6386-4. DOI: 10.1109/ANCS.2017.18. URL: https://doi.org/10.1109/ANCS.2017.18.

[100] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin. “IOT Gateway: BridgingWireless Sensor Networks into Internet of Things”. In: *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. Dec. 2010, pp. 347–352. DOI: 10.1109/EUC.2010.58.

[101] Yingdi Yu, Alexander Afanasyev, David Clark, kc claffy kc, Van Jacobson, and Lixia Zhang. “Schematizing Trust in Named Data Networking”. In: *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*. ACM-ICN ’15. San Francisco, California, USA: ACM, 2015, pp. 177–186. ISBN: 978-1-4503-3855-4. DOI: 10.1145/2810156.2810170. URL: http://doi.acm.org/10.1145/2810156.2810170.

[102] Michael Rambold, Holger Kasinger, Florian Lautenbacher, and Bernhard Bauer. “Towards Autonomic Service Discovery A Survey and Comparison”. In: *2009 IEEE International Conference on Services Computing*. Section II. IEEE, 2009, pp. 192–201. ISBN: 978-1-4244-5183-8. DOI: 10.1109/SCC.2009.59.

[103] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone, and L. Veltri. “A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things”. In: *IEEE Internet of Things Journal* 1.5 (Oct. 2014), pp. 508–521. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2358296.

[104] Badis Djamaa, Mark Richardson, Peter Barker, and Ian Owens. “Discovery of Things: A Fully-Distributed Opportunistic Approach”. In: *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)* (May 2015), pp. 1–5. DOI: 10.1109/VTCSpring.2015.7145778.

[105] Sameh Ben Fredj, Mathieu Boussard, Daniel Kofman, and Ludovic 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. ISBN: 9780769550466. DOI: 10.1109/GreenCom-iThings-CPSCom.2013.86.

[106] Sameh Ben Fredj and M Boussard. “Efficient semantic-based IoT service discovery mechanism for dynamic environments”. In: *Personal, Indoor, and ...* (2014), pp. 2088–2092.

[107] S. N. A. U. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim. “A unified semantic knowledge base for IoT”. In: *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. 2014, pp. 575–580. ISBN: VO -. DOI: 10.1109/WF-IoT.2014.6803232.

[108] Sejin Chun, Seungmin Seo, Byungkook Oh, and Kyong-Ho 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. ISBN: 978-1-4799-7935-6. DOI: 10.1109/ICOSC.2015.7050819.

[109] Clay Shirky. *Ontology is Overrated: Categories, Links, and Tags*. http://shirky.com/writings/ontology_overrated.html. Accessed: December 2015. May 2005.

[110] Gabriela Avram. “At the crossroads of knowledge management and social software”. In: *Electronic Journal of Knowledge Management* 4.1 (Jan. 2006), pp. 1–10.

[111] Thomas Gruber. “Ontology of Folksonomy: A Mash-up of Apples and Oranges”. In: *International Journal on Semantic Web and Information Systems* 3.2 (2007), pp. 1–11.

[112] Souleiman Hasan and Edward Curry. “Approximate Semantic Matching of Events for the Internet of Things”. In: *ACM Transactions on Internet Technology* 14.1 (Aug. 2014), pp. 1–23. ISSN: 15335399. DOI: 10.1145/2633684.

[113] Souleiman Hasan, Sean O’Riain, and Edward Curry. “Approximate semantic matching of heterogeneous events”. In: *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems - DEBS ’12*. New York, New York, USA: ACM Press, 2012, pp. 252–263. ISBN: 9781450313155. DOI: 10.1145/2335484.2335512.

[114] Glenn Scott. *CCNx 1.0 Simple Service Discovery.* Tech. rep. Computing Science Laboratory, Palo Alto Research Center, 2014.

[115] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T.C. Schmidt. "RIOT OS: Towards an OS for the Internet of Things". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on.* Apr. 2013, pp. 79–80. DOI: `10.1109/INFCOMW.2013.6970748`.

[116] M. Antunes, D. Gomes, and R. Aguiar. "Context storage for M2M scenarios". In: *Communications (ICC), 2014 IEEE International Conference on.* June 2014, pp. 3664–3669.

[117] Mário Antunes, Diogo Gomes, and Rui L. Aguiar. "Scalable semantic aware context storage". In: *Future Generation Computer Systems* 56 (2016), pp. 675–683. ISSN: 0167-739X. DOI: `http://dx.doi.org/10.1016/j.future.2015.09.008`.

[118] João Paulo Barraca, Diogo Gomes, and Rui L. Aguiar. "AMazING - Advanced Mobile wIreless playGrouNd". In: *Testbeds and Research Infrastructures. Development of Networks and Communities.* Ed. by Thomas Magedanz, Anastasius Gavras, NguyenHuu Thanh, and JeffryS. Chase. Vol. 46. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2011, pp. 219–230. ISBN: 978-3-642-17850-4. DOI: `10.1007/978-3-642-17851-1_18`.

[119] R. Ferreira, R. Aguiar, and A. Matos. "Recognizing entities across protocols with unified UUID discovery and asymmetric keys". In: *2013 IEEE Global Communications Conference (GLOBECOM).* Dec. 2013, pp. 2902–2908. DOI: `10.1109/GLOCOM.2013.6831515`.