Solving interoperability and performance challenges over heterogeneous IoT networks : DNS-based solutions

Antoine Bernard

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAS012

Thèse de doctorat

Solving interoperability and performance challenges over heterogeneous IoT networks – DNS-based solutions

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 26 Novembre 2021, par

M. ANTOINE BERNARD

Composition du Jury :

Laurent Toutain Professeur, École nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire	Rapporteur Président
André-Luc Beylot Professeur, École nationale supérieure d'électrotechnique, d'électronique, d'informatique, d'hydraulique et des télécommunications	Rapporteur
Monique Becker Professeur émérite, Télécom SudParis	Examinatrice
Philippe Cola Senior Core Network Architect, Bouygues Telecom	Examinateur
Ahmed Kamal Professor, Iowa State University	Examinateur
Michel Marot Professeur, Télécom SudParis	Directeur de thèse
Sandoche Balakrichenan Responsable recherche en partenariat, Association Française pour le Nommage Internet en Coopération	Co-encadrant de thèse
Benoit Ampeau Directeur Partenariat & Innovation, Association Française pour le Nommage Internet en Coopération	Invité Co-encadrant de thèse

INSTITUT POLYTECHNIQUE DE PARIS

Abstract

Télécom SudParis

Docteur de l'Institut Polytechnique de Paris

Solving interoperability and performance challenges over heterogeneous IoT networks – DNS-based solutions

by Antoine BERNARD

The Internet of Things (IoT) evolved from its theoretical possibility to connect anything and everything to an ever-increasing market of goods and services. Its underlying technologies diversified and IoT now encompasses various communication technologies ranging from short-range technologies as Bluetooth, medium-range technologies such as Zigbee and long-range technologies such as Long Range Wide Area Network.

IoT systems are usually built around closed, siloed infrastructures. Developing interoperability between these closed silos is crucial for IoT use-cases such as Smart Cities. Working on this subject at the application level is a first step that directly evolved from current practice regarding data collection and analysis in the context of the development of Big Data. However, building bridges at the network level would enable easier interconnection between infrastructures and facilitate seamless transitions between IoT technologies to improve coverage at low cost.

The Domain Name System (DNS) basically developed to translate human-friendly computer host-names on a network into their corresponding IP addresses is a known interoperability facilitator on the Internet. It is one of the oldest systems deployed on the Internet and was developed to support the Internet infrastructure's growth at the end of the 80s. Despite its old age, it remains a core service on the Internet and many changes from its initial specifications are still in progress, as proven by the increasing number of new suggestions to modify its standard.

DNS relies on simple principles, but its evolution since its first developments allowed to build complex systems using its many configuration possibilities. This thesis investigates possible improvements to IoT services and infrastructures. Our key problem can be formulated as follow: Can the DNS and its infrastructure serve as a good baseline to support IoT evolution as it accompanied the evolution of the Internet?

We address this question with three approaches. We begin by experimenting with a federated roaming model IoT networks exploiting the strengths of the DNS infrastructure and its security extensions to improve interoperability, end-to-end security and optimize back-end communications. Its goal is to propose seamless transitions between networks based on information stored on the DNS infrastructure. We explore the issues behind DNS and application response times, and how to limit its impact on constrained exchanges between end devices and radio gateways studying DNS prefetching scenarios in a city mobility context. Our second subject of interest consists of studying how DNS can be used to develop availability, interoperability and scalability in compression protocols for IoT. Furthermore, we experimented around compression paradigms and traffic minimization by implementing machine learning algorithms onto sensors and monitoring important system parameters, particularly transmission performance and energy efficiency.

INSTITUT POLYTECHNIQUE DE PARIS

Résumé

Télécom SudParis

Docteur de l'Institut Polytechnique de Paris

Contributions à la résolution de problèmes de performances et d'interopérabilité des réseaux IoT hétérogènes par l'utilisation du standard ouvert DNS et de services d'infrastructure

par Antoine BERNARD

L'Internet des Objets (IdO) a évolué depuis cette possibilité théorique de connecter tous les appareils à un réel marché de biens et de services en constante expansion. Les technologies sous-jacentes ont évolué et l'IdO repose aujourd'hui sur de nombreuses technologies de communication différentes: Des technologies à courte portée comme Bluetooth, moyenne portée comme Zigbee ou longue portée comme la technologie LoRa (Long-Range).

Les systèmes de l'IdO sont habituellement construits autour d'infrastructures fermées basées sur des systèmes en silo. Créer de l'interopérabilité entre ces silos fermés est un enjeu pour certains cas d'usages cruciaux dans le déploiement des technologies de l'IdO comme les villes intelligentes. Développer la problématique au niveau applicatif est une première étape directement inspirée des pratiques courantes en matière de collecte et d'analyse de données dans le cadre du développement des technologies de traitement de données massives. Cependant, construire des ponts au niveau réseau permettrait de faciliter l'interconnexion entre infrastructures et faciliterait la transition fluide entre technologies de l'IdO afin d'améliorer à bas coût la couverture réseau.

Le Système de Nom de Domaine (Domain Name System, DNS), initialement développé pour traduire les noms, lisibles et compréhensibles par les utilisateurs en adresses IP, utilisées par les appareils connectés, est reconnu comme un facilitateur sur les question d'interopérabilité sur Internet. C'est l'un des systèmes les plus anciens déployés sur Internet, développé à la fin des années 1980 pour supporter la croissance de l'infrastructures Internet. Bien qu'ayant beaucoup évolué ces dernières années, en témoignent les nombreuses propositions de modifications au standard publié à son sujet, le DNS reste aujourd'hui l'une des infrastructures les plus centrales du réseau Internet.

Le DNS repose sur des principes simples, mais son évolution depuis ses premiers développements ont permis de construire des systèmes complexes grâce à ses nombreuses possibilités de configuration. Dans le cadre cette thèse, qui étudie les possibles améliorations aux services et infrastructures de l'IdO, nous étudions la problématique suivante : Le DNS et son infrastructure peuvent-ils servir de support efficace à l'évolution de l'IdO de la même manière qu'il a accompagné l'évolution d'Internet ?

Dans cette optique, nous étudions de possibles améliorations de systèmes de l'IdO sous trois angles. Nous testons tout d'abord un modèle d'itinérance pour réseaux de l'Internet des Objets au travers de la construction d'une fédération reposant sur l'infrastructure du DNS et ses extensions pour en assurer l'interopérabilité, la sécurité de bout-en-bout et optimiser les communications entre infrastructures. Son objectif est de proposer des transitions fluides entre réseaux sur base d'informations stockées à l'aide de l'infrastructure DNS. Nous explorons également les problématiques introduites par le DNS, notamment en termes de latence et d'influence sur les temps de réponse des applications, et comment en limiter l'impact sur les échanges, déjà grandement contraints, entre objet connecté et passerelle radio. Pour cela nous étudions les conséquences de l'utilisation de requêtes DNS anticipées dans un contexte de mobilité en milieu urbain. Nous étudions ensuite la façon dont le Système de Nom de Domaine peut renforcer l'interopérabilité, la disponibilité de ressources et le passage à l'échelle de systèmes de compression de paquets de l'IdO. Enfin, nous explorons la question de la minimisation de trafic en implantant des algorithmes d'apprentissage sur des capteurs et en mesurant les paramètres du système final, en particulier en terme de performances de transmissions et d'efficacité énergétique.

Acknowledgements

Looking back on this adventure, now comes the time to thank all those who believed in me and accompanied me for these three years.

First of all, I would like to express my sincere gratitude to my director Michel Marot, who followed this work as closely as possible. I was fortunate to have the opportunity to work with him. I learned a lot from our discussions; I won't forget his advice for the years to come.

I also thank Sandoche Balakrichenan, who supervised my day-to-day work within Afnic, pushed me to move forward with my work, challenged me about the propositions I made and helped improve them with his knowledgeable comments.

My thanks also go to Benoit Ampeau, our boss at Afnic, who accompanied the execution of our projects, offered his expertise when necessary and contributed to the valorization of our experiments and results. And I thank the Afnic, and its director Pierre Bonis, for financing my work for these three years.

I thank Laurent Toutain, whose work was a real inspiration for this thesis, who accepted the task to review it and who presided the defense jury. I also thank him for his comments to improve the present manuscript.

I also thank André-Luc Beylot, who agreed to preside my mini-defence and shoulder this manuscript's review. In both these instances, his advice and critics helped me improve.

My thanks also go to Philippe Cola and Ahmed Kamal, who agreed to participate to my defence as jury. And to Monique Becker, who, on top of her participation in the jury, put her experience to use by reviewing my work.

My next thanks are addressed to all my colleagues from Afnic: Pierre-Aymeric Massé, for his advice at the beginning of this adventure, Hélène Boudon, Alexandre Pion and Gaël Berthaud-Müller for our discussions, not always related to work but always insightful, Michal Toma, Samia Mtimet, Lotfi Benyelles and Lucien Castex for their feedbacks, Regis Massé for following our work and offering to review and comment it when necessary and Stéphane Bortzmeyer for his insight and well of knowledge regarding DNS and all IETF-related topics.

I also thank all my colleagues from Telecom SudParis, namely Aicha Dridi and Mohamed Laroui, for our joint work in our shared office, Hossam Afifi for his advice, his insight and his answers when working together on machine learning, back when I was a complete novice on the subject, Vincent Gauthier for his insight and advice that taught me a lot about possible solutions to the issues I encountered, Hassine Moungla for his advice on how to valorize our work. I thank Bruno Defudes, who was tasked to review my first year's work and check that everything was doing fine and Hind Castel, who accompanied André-Luc in my mini-defence. My thanks to Véronique Guy, Ydalia Garcia, Sandra Gchweinder and Valérie Mattheus, each for accompanying my work for these years, be it assisting with conference registration or for accompanying my teachings. Speaking about teachings, I would also like to thank Olivier Paul for accepting me as a part of his networking course's teachers. I

also thank Cedric Adjih and Alexandre Abadie for their insight when discovering IoT technologies.

The remaining people to thank are, of course, my friends: Adrien, Paul, Victor, Erwan, Pierre, Fabien, Lauriane, Mockie and Mad, to acknowledge a few, for so much but I'll only mention our games periods, behind a computer screen or around a table. A thought to all my former colleagues, schoolmates and teachers, with special thanks to Dominique Chiaroni for introducing me to research through my internship with him at Nokia Bell Labs and to MiNET's members, past, present or future, who are doing formidable work and with whom I discovered networking.

Last but not least, I would like to thank my family: my grandparents, parents and brothers for supporting me throughout all my life or theirs and Clara, who supported me during our, already, four years together.

This work was partly financed by the French National Research Agency through the CIFRE program [2018/0668].

This work benefited from the support of the Energy4Climate Interdisciplinary Center (E4C) of IP Paris and Ecole des Ponts ParisTech. It was supported by 3rd Programme d'Investissements d'Avenir [ANR-18-EUR-0006-02].

Contents

List of Figures

List of Tables

List of Abbreviations

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
AA	Authentication and Authorization
AAA	Authentication, Authorization and Accounting
API	Application Programming Interface
AS	Application Server
CA	Certificate Authority
CoAP	Constrained Application Protocol
DANE	DNS Authentication of Named Entities
DNS	Domain Name System/ Domain Name Service
DNS-SD	DNS-Based Service Discovery (RFC 6763)
DNSSEC	DNS Security Extensions
Do53	DNS over UDP (port 53)
DoH	DNS over HTTPS
DoQ	DNS over QUIC
DoT	DNS over TLS
EAP	Extensible Authentication Protocol
ED	End Device
EPC	Electronic Product Code
EUI	Extended Unique Identifier (aka MAC address)
fNS	forwarding Network Server
HN	Home Network
HTTP(S)	(Secured) HyperText Transfer Protocol
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IoT	Internet of Things
ISO	International Organization for Standardization
JR	Join Request
JS	Join Server
LoRa	Long-Range
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LSTM	Long Short Term Memory
LTE-M	Long Term Evolution - cat M1
M2M	Machine-to-Machine
MAPE	Mean Absolute Percentage Error
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrow-Band IoT
NS	Network Server
ONS	Object Naming Service
OS	Operating System
OTAA	Over The Air Activation
OUI	Organizational Unique Identifier

PSK	Pre-Shared Key
RDF	Resource Description Framework
RFC	Request For Comments
RFID	Radio-Frequency IDentification
RG	Radio Gateway
RNN	Recurrent Neural Network
RR	DNS Resource Record
RTT	Round Trip Time
SCHC	Generic Framework for Static Context Header Compression and Fragmentation
SDO	Standards Developing Organization
SF	Spreading Factor
sNS	serving Network Server
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
TCP	Transmission Control Protocol
TFLite	TensorFlow Lite
TLS	Transport Layer Security
U/D-RTT	Uplink/Downlink Round Trip Time
UDDI	Universal Description Discovery and Integration
UDID	Unique Device IDentifier
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VN	Visited Network
WBA	Wireless Broadband Alliance
WSN	Wireless Sensor Network

Sur des épaules de géants

Chapter 1

Introduction

From the early development of the Internet, identifying machines inside the network has been a necessity. One of the first systems to identify and design machines on the network was a single file (called the host.txt file) shared across the network, which contained all the identifier/name tuples. As the number of machines grew, the identification system moved away from this single shared file to the newly developed Domain Name System (DNS).

The DNS is a hierarchical system that allows users to generate names for their devices on the network. DNS is often presented as the phone book for the Internet, allowing users to remember domains, URLs and email addresses instead of the corresponding IP addresses. With its roots spread in various locations, the DNS tree proved to be a resilient distributed system, easy to access from anywhere on the planet and support interoperability and services discovery on the network.

The DNS is a massively deployed and scalable system supporting worldwide communications on the Internet. Its deployment was a valuable tool to accompany the increase in the number of machines on the network. Additionally, the ease of use and registration of domain names helped create new markets on the Internet, one of which was the popularization of Web technologies. The system has many uses and is regularly improved by continuous development and standardization efforts.

The DNS is sometimes exploited to develop new uses such as Apple's Hello, proposed for standardization as DNS Service Discovery (DNS-SD). This sort of hacking to the DNS principal goal, exploiting its main characteristics (interoperability, global repartition) to create new uses of the DNS paradigms and architecture, is precisely the kind of possible uses we aim to develop in this thesis.

This thesis aims to study and develop such transverse uses of DNS regarding the Internet of Things (IoT) and its infrastructure. We hypothesize that the DNS might prove a useful tool as it would permit developing new IoT uses without specific overcost; that the DNS might improve solutions linked to IoT and help build a Web of Things through registries containing devices information. Building such a system should help IoT evolve the same way as the Internet, building interoperable systems and networks backed by cloud servers sharing actions and data. Moreover, reusing existing technology such as the DNS should prove an efficient way to accompany IoT integration toward the rest of the Internet.

This thesis focuses on wireless communication in Wide Area Networks.

Most IoT, among those Low Power Wide Area Networks (LPWANs), systems are now built as incompatible isolated data silos with each technology transmitting over

its waves, using its infrastructure and centralizing data in data warehouses for exploitation. Nowadays, around 99% of these Things are not connected to the Internet [1]. Using the DNS to break data and communication silos would allow infrastructures and devices to communicate freely and build interoperable networks between IoT technologies.

Another key similarity between the Internet and the Internet of Things comes from the similarity in their problems. IoT is riddled with scalability, interoperability, mobility and roaming, transmission efficiency, availability, reliability and other security issues such as privacy. The DNS contributes to solving many of these issues on the Internet, hence our interrogation on possible improvements to IoT systems backed by the DNS infrastructure.

This thesis studies IoT systems regarding the following key aspects: Naming, Roaming, Header Compression and Payload Compression. Though security aspects will be detailed in various part of the thesis, security weaknesses of IoT End Devices (EDs) is outside the scope of this thesis.

We focused our studies on LPWANs; LPWANs are constrained networks built, in general, as star typologies supporting a massive quantity of EDs around a single antenna. Among these LPWAN, our work focuses on Long-Range Wide Area Network (LoRaWAN) as the network is open and easy to access, provides interesting constraints but necessitate no additional cost. It is easy to build LoRaWAN networks and participate in the evolution of the specifications.

1.1 Research issues

IoT systems are usually built around closed, siloed infrastructures. Interconnecting such silos at the network level is not an easy task, and, as LPWANs are usually used to gather data from sensors, it is usually more straightforward for a user to build Application Programming Interface (API) for each application based on its needs than to break the network silos.

Breaking the silos is a key subject to opening the IoT Infrastructures and bringing the IoT work to the global Internet. Breaking them at the applicative level is a first step that directly evolves from current practice regarding data collection and analysis in the context of the development of Big Data. However, breaking the silos on a network level would permit easier interconnection between infrastructures and help to build seamless transitions between IoT technologies to improve coverage at low cost.

DNS is a known candidate to break such a silo. DNS relies on simple principles but its evolutions since its first development allows us to build complex systems using its many configuration possibilities. Possible uses of DNS to develop IoT systems are not extensively studied for the following reasons:

- Most IoT research studies perceive IoT as a separate network from the Internet. Hence existing Internet protocols are not a research focus for IoT developments.
- DNS is usually developed in Standards Developing Organization (SDO) and DNS extensions are seen as operational issues (academic work on DNS is often performances or security issues/solutions);

- DNS which has been conceived for the Internet is considered not to be appropriate for constrained IoT requirements.

DNS is a naming service wherein the basic use of DNS is to map the identifier to its specific service over heterogeneous applications/services in the decentralized Internet. Since the IoT systems are mostly siloed and centralized, the usage of DNS as a naming service is still not well studied. Based on these observations, studying DNS through academic work as a possible tool to break IoT silos and support IoT infrastructures seems an interesting subject. Considering our third requirement described above, our goal is not to embark DNS protocols onto sensors but instead use DNS on the infrastructure side to support IoT improvements. Proposing such improvements is our first goal.

The IoT research community would also benefit from additional experimental work with IoT systems to verify the experimental feasibility of its solutions. Thus our second goal is to run experiments to test hypothesis for IoT based on DNS. This experimental approach introduces additional constraints such as working with reference implementations of the solutions, generating actual IoT traffic for measurements and analysis, respecting airtime constraints or device lifecycle.

1.2 Motivation

1.2.1 IoT Roaming

Roaming requires an interconnection agreement between network operators. Interconnection in IoT becomes possible by establishing a direct 'One-to-One' interconnection or building an interconnection 'Hub' for operators. Establishing an interconnection agreement with a single hub makes it possible to exchange traffic with the other operators connected to that hub. Both the hub and the One-to-One interconnection models evolve as independent Silos wherein the device in the coverage area of a Visited Network (VN) can connect to its service only if there is a prior interconnection agreement between its Home Network (HN) and the VN or between the HN and the interconnection hub.

The interconnection agreement serves as a basis for mutual authentication and authorization between the operators. In the 'One-to-One' interconnection model, bootstrapping trust is a key security concern. The ED needs to be cryptographically authenticated by the VN based on credentials such as its identifier and a Pre-Shared Key (PSK). Cryptography-based authentication usually relies on one or more trust anchors [2]. In the proprietary silo scenarios, the trust anchor information may be preset on the ED or established out of band.

Also, interconnection agreements define how operators should communicate between themselves, including setting up mutual authentication between backend elements to secure backend exchanges on the network. In the 'Hub' scenario, the operators are asked to ultimately trust the centralized hub, whereas, in a 'One-to-One' model, the agreement usually defines the specific backend authentication and authorization method to be used between the operators.

The goal of the first part of this thesis is to address these issues by proposing an architecture, IoTRoam, federating different organizations to allow flexible, mutual authentication and authorization between any backend element in a roaming situation, without a direct and explicit roaming agreement (e.g. for the LoRaWAN case,

interconnection of Network, Application and Join Servers between operators). The agreement is implicitly given when the organization joins the IoTRoam federation.

Moreover, any architecture proposing solutions to the technological barriers mentioned earlier should consider the constrained characteristics of IoT environments. LoRaWAN is one of the most constrained IoT networks, having heavy constraints on payload size and latency. Any roaming approach for LoRaWAN is expected to obtain satisfying results with other IoT networks. LoRaWAN mainly consists of various siloed infrastructures deployed by small operators with no prior agreements and common configuration between them. Furtherly, LoRaWAN, as a single connectivity solution among the LPWANs, is a communication silo as communicating between a LoRaWAN ED and a SigFox ED needs specific backend development and usually consists of exchanging data through a broker. We can even say that LoRaWAN is a single connectivity solution among all IoT instead of LPWANs and that communicating between a LoRaWAN ED and a WiFi ED requires specific interfaces or connectivity between backend through a data broker. Thus we designed and experimented with our proposed method as a LoRaWAN interconnection architecture as a benchmark.

Our approach to building our roaming architecture was to use the combination of the DNS infrastructure and a PKI to build a secure open roaming infrastructure accessible to public and private LoRaWAN operators. The possibility to set up private networks for free is a competitive advantage of LoRaWAN. We designed, built and deployed a proof of concept architecture to test the Roaming capabilities offered by the ChirpStack solution and test roaming between private and public LoRaWAN. We also validated our infrastructure by testing LoRaWAN connectivity for EDs in a roaming context by studying various onboarding scenarios, measuring onboarding time and communication delays.

Complementary, we study roaming data prefetching in a multi-tenant scenario through simulation based on device mobility at the scale of a megapolis, Rome. This permits us to experiment on predicting movements to increase prefetching efficiency. We simulate an antenna placement and design DNS caching and prefetching scenarios based on vehicle movement traces. Antennas activation and DNS queries handling is studied to understand differences between handling data from querying the DNS when needed or prefetching information in a localized cache to reduce on-the-fly latency when the information is actually necessary.

Combining an ML predictor and a prefetching mechanism would reduce eventual latency introduced by the DNS while soliciting a limited amoun of antennas within the movement perimeter. Exploiting this solution, usually used by web browsers to reduce latency, is an interesting topic to address in a different scenario with such a different use case.

1.2.2 Header Compression

Compression is a well-studied subject, be it when compressing images or files. However, compressing communications is still an evolving aspect of the compression research. Let us first consider header compression, the case of payload compression being addressed later. One such compression technology is Robust Header Compression (ROHC), which compress data based on redundancy between packets in a given flow. Another is SCHC: Generic Framework for Static Context Header Compression and Fragmentation [3]. SCHC is a recent standard developed by the

Internet Engineering Task Force (IETF) to compress, decompress, fragment and reassemble packets transmitted over LPWANs.

Compressing data using SCHC relies on realizing a pattern matching on the packet header before transmitting it over a constrained network, from the ED to its backend or from the Radio Gateway (RG) to the ED. In order to compress the data sent and received between the ED and the backend, SCHC uses a predefined group of rules called Context, which is deployed on both the ED and on the backend.

For every Context, there could be a single or multiple rules. When sending data from the ED to its RG, the SCHC Context rule enables compression by suppressing redundant, superficial, predictable or most used data inside an IPv6 header and replacing them with a Rule Identifier chosen in a given set of predefined rules. For instance, the ED's IP address may be added to the Context allowing it to avoid transmitting 128 bits IP address data if all the packets sent by a sensor have the same IP address.

When using SCHC, one element from the backend should realize SCHC operation (compression, decompression, fragmentation, reassembly) for all associated EDs. That is why we propose that the RG, the Network Server (NS) or the Application Server (AS) retrieve the Context dynamically from a remote server. Thus, the owner of the rules could easily modify them. Only the Rule IDs and versions are stored in either the RG, the NS or the AS.

Also, storing all SCHC rules, considering they might be unique for each ED, might introduce scalability issues to the system. We can consider around 20 rules per ED when working with such rules (as the rule ID + rule length is encoded on 1 byte), with thousands of EDs around a single antenna and multiple antennas for a given server (as LoRaWAN is built as a star of stars topology). When considering hundreds of thousands of EDs around a single server with up to 10kb per Context rule, we end up storing gigabytes of data to enable SCHC on a given LoRaWAN infrastructure.

At last, the current way to store the SCHC Context rules statically does not allow to roam easily. It lacks the necessary flexibility which would enable to development of roaming capabilities when using SCHC. The use of an Administration Management System (AMS) as proposed by [4] could be a solution. It is a proposition to supervise roaming agreements and manage the use, generation and exchange of SCHC Contexts. However, such roaming accords would prove experimentally tricky when working with multiple operators, thus building accords between each LoRaWAN AMS, considering that operating a LoRaWAN network is possible at low cost without paying licenses for emitting data over the air.

There are multiple options for storing these Context rules. For example, it could be done in a private server, stored in the cloud or directly embedded within the operator's server. However, we think that it could be wise to use an open, distributed mechanism to find the location of the server where the Context rules are stored. We propose to experiment possible use of DNS as a way to support SCHC compression/decompression mechanism. As an optimized, hierarchical and distributed database, DNS could enable identifying the server's location where the Context rules are stored feasibly on the Internet. Hopefully, using such a mechanism would allow for a seamless transition, from preconfiguring the information needed on the backend to building it dynamically, on the fly, based on actual needs when operating the network.

DNS would prove an efficient solution to introduce more flexibility and improve scalability when using SCHC. Our solution would provide open access to SCHC parameters as a way to support roaming capabilities. Considering these three aspects of the SCHC framework, namely improving its flexibility, scalability, and assisting SCHC when an ED is roaming, and how DNS might help, we ask ourselves if it is possible to host rules outside the scope of the ED's NS without hindering the transmissions. To solve this problem, we deployed a dynamic Context resolution architecture based on DNS for SCHC compression/decompression and studied the consequences of such mechanism on the system latency and other possible consequences on LoRaWAN communications.

1.2.3 Data compression

After our work on compressing headers in LPWAN, we decide to further our approach regarding transmission efficiency by studying payload compressing. The easiest way to reduce data transmission is to delete redundancies or to round them to near values. When working with sensors, data is often time-correlated. For example, the temperature may vary slowly. Recently, Neural network-based techniques entered the landscape of IoT data compression techniques. Data can be compressed by their regression curve inferred from a neural network. More complex prediction methods can also be used. Neural networks are known as universal function approximators with the capability to learn arbitrarily complex mappings, and in practice, show excellent performance in prediction tasks. Other approaches include the use of Long Short-Term Memory networks (LSTMs) to perform predictions.

Many compression techniques appeared over the years. Nevertheless, if the "classical" methods present efficient compression ratios, they do not avoid transmitting data. Periodically a sensor senses data, may compress it and then send the compressed payload, but compressed data payload (plus header) are still sent. New neural network-based techniques appeared, and they avoid sending data in situations where the prediction is good. A neural network-based predictor is implemented in the ED and also at the back end. If the sensed data is well predicted, no data is sent, and the backend uses the prediction. Otherwise, the measurement is sent.

These approaches raise several questions. First of all, they have not been tested in real experiments yet. We like to push theoretical subjects further by implementing solutions directly on EDs. Thus we aim to experiment with an actual LSTM implementation on sensors to back or disprove the results obtained through simulations by the scientific community.

An important issue has not been addressed in the literature: questions on the complexity of a given Machine Learning (ML) model considering its integration to constrained devices lead us to study the efficiency of a given neural network regarding its size and the possible consequences of embarking such an algorithm directly on devices. It is commonly admitted that ML algorithms are too heavy for the constrained devices which are the sensors: their memories and processing capabilities are too small and their battery requirements too strong. Even if machine-learning algorithms may be embedded on a device, would its energy consumption be compatible with the small battery and the requirement to be alive for years? At last, there are coding issues related to the precision of the numbers processed by the ED: the number of digits is often limited in such a device, leading to weights quantization. Also, the number of weights of a neural network may be huge, and it makes

sense to try minimizing the memory size by shortening the number of digits the weights are coded with. In this case, what is the impact of weights quantization on the algorithm's accuracy, and in fine the compression efficiency?

Lastly, we aim to study possible uses of the DNS infrastructure to back such compression mechanism by embarking the weights directly on the DNS. With DNS being an efficient open distributed database, using DNS would permit to open the access to ML models and weights, for example, when an ED is roaming. Three scenarios are studied about publishing ML weights in the DNS, each with its own strengths and weaknesses:

- one of them relies on asking the backend to publish the ML weights in the DNS.
- a second consists is to rely on the DNS to store the address of a given API which would allow to retrieve and modify ML algorithm
- a last possibility consists of learning the lessons from the DNS-SD paradigm [5] in a dual connectivity infrastructure to support discovery and advertising of ML model within a given local coverage, for example, as a way to support compressed LoRa Mesh communication.

1.3 Contributions

Our work mainly consists in breaking the silo between multiple LoRaWAN deployments, but as mentioned above, we expect similar results when working with less constrained networks.

Our work with LoRaWAN led us to collaborate with various parties from the community and contribute to the LoRaWAN Specifications, and collaborate with the lead developer of the ChirpStack open-source software, which is the most massively used and reference LoRaWAN infrastructure backend solution. Our focus when working with ChirpStack mostly consisted of providing insight on the interconnection between the ChirpStack Solution and the DNS infrastructure regarding the actual use of DNS in the LoRaWAN specifications.

This work also leads us to reflect on a possible use of DNS prefetching to reduce the impact from usual DNS querying in a mobility use case. When querying information necessary for device's functionalities, prefetching the necessary information beforehand allows interesting improvements in terms of DNS cache hit. We provide simulation results based on a comparison between three querying scenarios based on mobility traces in an urban area.

After extending roaming capabilities in a LoRaWAN network, we worked on compressing headers to develop IP connectivity in LPWANs. Our contribution relies on extending the existing SCHC standard by leveraging the DNS infrastructure in its capability to increase connectivity and accompany application scalability.

Thus we proposed an experiment to measure the time delay induced by using the SCHC compression/decompression framework, then extended the time delay study by adding new interfaces to SCHC, enabling remote context querying. We also provide experimental measurements on DNS querying time to improve the quality of our time measurements by studying actual resolution time outside our lab scope.

Finally, we extend our work on compression by working on compressing payload using ML techniques and studied possible strengths and weaknesses of such system.

1.4 Structure of the thesis

This thesis is structured as follow.

Chapter 2 presents a state of the Art regarding naming, and more specifically and IoT. It first presents the scope for identification in the IoT ecosystem, then presents a few aspects on querying and its underlying mechanisms (architecture, performances, methodology). After that, scalability and security are presented with a focus on the specific aspects that are of interest for the thesis. Coverage and roaming as solutions to support mobility are described as well as the DNS, its functioning and evolutions. Finally, we provide references on various approaches to combine ML paradigms and IoT.

Chapter 3 presents our work on roaming and how DNS improves roaming capabilities and break IoT silos. The work is first introduced in 3.1 and compared to other approaches (3.2). Then we provide insight on LoRaWAN with regards to interconnecting networks (3.3) and our focus and design choices with regard to identification (3.4) and security integration (3.5). After that, we dig into our experiment and present our setup (3.6) and measured performances (3.7). We propose to extend this solution by provisionning the DNS queried information beforehand using a combination of prefetching techniques and mobility prediction algorithms (3.8). Finally, we sum up our contributions (3.9), provide our conclusions (3.10) that summarize the work and propose future steps for the experiments.

Chapter 4 describes our work on SCHC and its extension using the DNS infrastructure to support rules management. It starts with a short presentation on SCHC (4.1) before digging into the experiment (4.2) in which our propositions, scenarios and testbed are described. Then our results are described (4.3) and discussed (4.4).

Chapter 5 presents our implementation of LSTM on constrained devices using our own development working with pre-calculated weights from TFLite. An introduction centers the subject around our usual scope (5.1) and put it into perspective with usual compression techniques (5.2). Then we describe our experimental approach (5.3) and discuss our results regarding our various measurements points (5.4) and weight storage modalities (5.5). The chapter ends with a short conclusion (5.6) that summarize our contributions and describe possible further approaches to the subject

Finally, the last chapter (Chapter 6) concludes the thesis by providing an overview of our contributions and presents possible extensions to the work we realized.

Chapter 2

State of the Art

2.1 IoT Scope

The term "Internet of Things" (IoT) encompasses several meanings depending on the communities/technologies involved. The basic purpose is to connect the "Things" in the physical world to the Internet infrastructure. The things could be anything from computers to people to medicines to books.

The things could be connected to the Internet infrastructure directly or indirectly. A Computer or a mobile phone could be connected to the Internet directly using an IP stack and some type of layer-2 connectivity, such as Wi-Fi or Ethernet. People or books will have indirect connections to the Internet, which may be enabled via some intermediate equipment, typically a non-IP carrier device, such as sensors, Radio Frequency Identification (RFID) or Near Field Communication (NFC), tagged with the things.

These carrier devices do not use the Internet protocol suite (TCP/IP) for communication. Instead, they use their proper communication technologies such as Bluetooth, Zigbee or Long-Range (LoRa). To link the non-IP-capable devices to the IP network (i.e. the Internet), there is a need for a gateway device, which can handle communication at two levels: on the one hand with the non-IP-capable devices, and on the other hand, with the IP network; thus bridging between non-IP and IP worlds.

2.1.1 Making "things" "smart"

The basic idea for IoT is to make the "things" "smart", which are otherwise considered dumb by default, from a technical perspective. Let us take the example of a cow in a herd, which is an entity of interest (Figure 2.1) for the farmer. Every 21 days, the cow has a 12 to 18 hours window, which is considered as the optimum period for mating [6]. The cow is highly active during this window, and hence the IoT application is attaching pedometers to the cow. The pedometer tagged to the cow periodically sends information, and a message is triggered to be sent to the farmer when the cow is walking more than its usual average. There are such a plethora of applications where things in the physical world can be tagged to make it smarter.

The progress in hardware development, decline of size, cost and energy consumption has enabled the feasibility of tagging non-IP devices to the physical things. This is why there is much talk about IoT currently, even though the idea is not new.

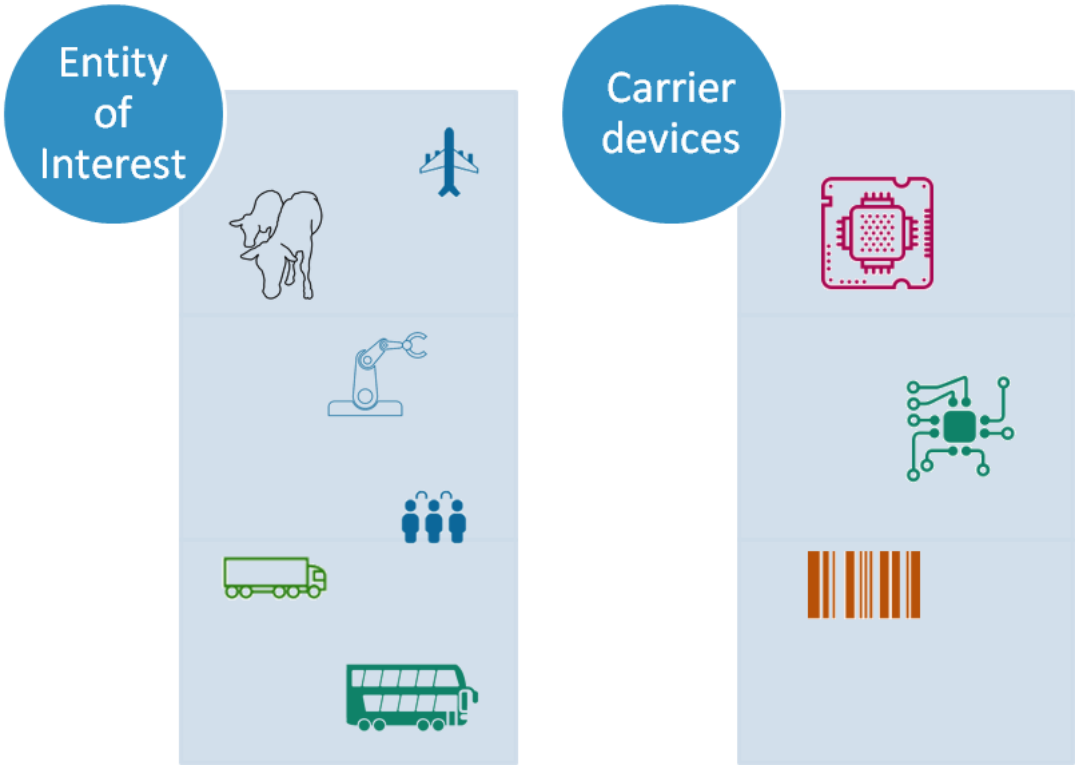


FIGURE 2.1: Making the things smart by tagging carrier devices such as sensors, RF-ID, barcode

2.1.2 Identifying "things"

Taking the cow example described previously, the farmer needs to identify a cow individually in his herd. For this purpose, the pedometer tagged to each cow in the herd should have a unique identifier. The scope for the uniqueness of the identifiers is limited within the herd.

However, IoT envisions billions of devices connected to the Internet. Hence, the identifier for each thing should be unique in the IoT. In the current Internet infrastructure, identifying a thing (a computer or a router is also a thing from an IoT perspective) uniquely on the Internet is based on IP addresses (either IPv4 or IPv6). The IP addresses follow a specific naming convention [7]. There is a hierarchical structure [8] which provisions the IP address and makes sure that there is no duplicity (i.e. no two devices on the Internet has the same IP address). Some other things may not use global IP addressing, using private addressing instead. Things having a private address are still connected to the Internet, with the help of a gateway device, which uses a global IP address to transport data from a private network to the Internet and vice versa.

As mentioned earlier, IoT involves non-IP capable devices; hence they do not use IP addresses for identification. The way these devices are identified could be classified into legacy and emerging identification. The legacy identifiers have their existing naming conventions, their proper structure to provision their identifiers to end-users, well before the emergence of the IoT theme. These legacy identifiers range from EUI-48, EUI-64 for MAC addresses (Extended Unique Identifier), Digital Object Identifiers (DOI) for electronic content, and Electronic Product Code (EPC) for RFID or barcodes. The emerging ones are new naming conventions with their proper

provisioning structure, developed to satisfy specific needs of a particular section of the IoT industry.

Identification plays a vital role in interconnecting heterogeneous IoT networks. When an IoT ED is roaming, the VN should retrieve its identifier and bootstrap the interconnection process to access the service related to the identifier on the Internet. If the obtained identifier is not unique, then there is a possibility of collision. Hence the identifier for an IoT ED should be unique.

Identifiers used in IoT includes heterogeneous identifiers encoded in different standardized naming formats such as IPv6, EUI-48, EUI-64, EPC, DOI, RF-ID, non-standardized identifiers for a specific industry such as Apple Unique Device Identifier (UDID) and user-generated identifiers.

One possible way to solve the issue of heterogeneity in identifiers is for all IoT stakeholders to move to a globally unique identification or addressing scheme, identifying using IPv6 addresses might prove to be a good solution (since it has a large addressing space and is capable of allocating a unique identifier for every IoT ED on the planet). In reality, migrating to one globally unique identifier for the whole IoT industry is impossible. The reason being cost and the technical complexities in migrating the IoT infrastructure with their existing identifiers to IPv6. A feasible alternative would be to let the different sectors in the IoT use their existing identifiers but to use a mapping service, which can map the identifier of an IoT ED to its network.

The well-known mapping service on the Internet is the DNS [9] [10], basically conceived to translate human-friendly computer host-names on a TCP/IP network into their corresponding "machine-friendly" IP addresses. Afnic's previous work [11] provided arguments based on existing standards and deployments to illustrate why DNS should be the naming (i.e. mapping) service for IoT.

Examples of leveraging DNS for mapping identifiers other than domain names include E.164 Number to URI Mapping (ENUM [12]) for telephone numbers. For IoT, there exists already standards such as Object Naming Service (ONS) [13] for the consumer industry, Object Resolution System (ORS) standardized jointly by the ITU-T and ISO/IEC and the Handle system standardized by the ISO, which uses the DNS infrastructure to resolve the IoT identifiers to its related service on the Internet.

IoT identifiers are structured into two different categories: Hierarchical and Flat. The hierarchical identifiers are allocated hierarchically, control is decentralized, and the nature of allocation ensures no duplicity exists. These features are similar to the domain name allocation and management, and thus these types of identifiers could naturally leverage the DNS infrastructure for allocation and resolution.

An example of a hierarchical identifier used in the supply chain industry is the EPC. The barcodes attached to consumer products follow the EPC naming convention, which can be hierarchically partitioned into Country, Organisation and product levels.

An example of a flat identifier is UDID, a unique serial number assigned to each Apple manufactured device. Apple uses UDID to track and record Apple manufactured devices, and it does not have a hierarchical allocation as that of EPC. The UDID is unique within the Apple UDID namespace. It is a 40-character alphanumeric string of code as follows:

2b6f0cc904d137be2e1730235f5664094b831186

Provisioning both identifiers types; EPC and UDID could be included into the Internet via the DNS namespace. Then it is up to the client libraries to make the conversion and add the specific sub-domain suffix ("udid.apple" for UDID and "gs1" for EPC) to the identifiers. Once the identifier is converted to a fully qualified domain name as follows:

2b6f0cc904d137be2e1730235f5664094b831186.udid.apple.
3.1.3.1.6.2.3.3.9.3.4.0.3.gs1. (supposing that there is a TLD called 'gs1')

They will follow the standard DNS resolution to resolve their associated resource, ED or metadata.

We hypothesize is that DNS is the only infrastructure that could scale to billions of EDs in the context of IoT interconnection, similar to how it has withstood the meteoric rise from hundreds of domains at the beginning of the Internet to billions currently [14]. Thus, we propose using DNS infrastructure as a scalable solution to satisfy the requirements outlined by Wireless Broadband Alliance (WBA)[15].

2.1.3 Overcoming interoperability challenges

Interoperability is the ability of a system to work with or use the components of another system. As mentioned in the article [16], there are four levels of interoperability; we will narrow our focus on an Organizational Interoperability approach to overcome interoperability challenges. Organizational Interoperability is the ability of organizations to communicate and transfer information across different information systems effectively, infrastructures spanned over different geographic regions, and cultures [17]. The European project - symbIoTe [18], proposes a finer granularity of Organizational Interoperability, enabling IoT platforms to collaborate by forming federations, thus supporting roaming where the EDs could find their core services while in the coverage area of the VN with the help of their unique identity.

[19] points our the necessity of adopting a standardization approach to improve IoT technologies. Without such approach, the IoT ecosystem would end up fragmented between specific technologies and use cases. Another key aspect pointed out is the importance of improving interoperability between IoT solutions and the necessity of a clear legislative framework to ensure the right to privacy and improve security for all users.

2.2 Querying

Querying in IoT raises many research issues: data storage and database comparison, temporal and spatial retrieval, the architecture of IoT, data and query distribution (and, subsequently, caching mechanisms), interoperability, data contextualization (and what is related that is discovery), and query performance. Concerning the used methods, they are numerous, for example, probabilistic ones, machine-learning-based ones or, of course, graph-based ones and blockchains.

2.2.1 Data storage

[20] compares relational and graph databases. [21] compares SQL and NoSQL databases for a small scale IoT application of water sprinkler system and investigates whether NoSQL performs better than SQL in different scenarios. NoSQL is the database technology that allows the rapid organization of different non-relational data types. NoSQL databases are divided into four categories, which are key-value based, column-oriented, document-oriented and graph databases. In [22], it is aimed to evaluate the storage and query performance of MongoDB on IoT data. The authors of [23] address the issue of data storing with a big data approach using MongoDB to spread data over servers and maximize query speed uniformly. Resource Description Framework (RDF) is a graph-based data model employed for representing the Uniform Resource Identifiers (URIs), and SPARQL is the standard query language used for processing the query of RDF data. The growth of data throws a big challenge to data storing and processing. In [24] a new data storing and query processing approach is proposed using a weighted predicate tree. The predicate tree is used to effectively store data and extract the weights indicating the relation of the data. [25] devises a SPARQL query engine able to factorize on-demand and semantically enrich stream data in a knowledge graph. The problem with ontologies to describe concepts, relationships between entities in various application domains is that semantic technics increase the complexity. Based on RDF as data format and SPARQL as a query language, [26] propose an in-network query processor to face the challenge of handling verbose RDF data and SPARQL queries execution in embedded devices. Most XML (eXtensible Markup Language) data are indexed by partitioning them into several data streams, which results in processing multiple data streams simultaneously when querying, and it is heavy. [27] uses a novel index storing only a subset of the data nodes, the rest of the nodes being generated efficiently and its authors propose an algorithm to process one data stream at a time.

To eliminate the adverse effects on massive data processing in IoT, a novel skyline preference query strategy based on massive and incomplete data set is proposed in [28]. [29] presents a single-node datastore able to ingest multidimensional sensor data at very high rates. Its design centers around a two-level indexing structure, wherein the global index is an in-memory R*-tree and the local indices are serialized k-d trees. [30] designs a memory-efficient high-performance key-value system called RadixKV, which offers efficient improvements on the Radix Tree. [31] proposes a lightweight and secure IoT remote monitoring mechanism using DNS with privacy preservation. The communication between IoT devices and gateways uses conventional protocols such as Constrained Application Protocol (CoAP) and Message Queuing Telemetry Transport (MQTT), while only remote monitoring uses DNS protocol. That is encrypted IoT data, after being encoded with base64, is stored as a DNS TXT record of the domain name of the IoT device and only the designated users are allowed to query and decrypt the data based on TSIG (Transaction SIGnature) authentication of DNS protocol and asymmetric cryptography.

2.2.2 Temporal and spatial retrieval

Time-series

In order to exploit and analyze the time-series data efficiently, [32] develop an index and the matching approach for the continuous time-series data, contrary to most of the static indexing approaches. Its authors propose a lightweight index

structure, L-index, and a matching approach, L-match, for the constraint normalized subsequence matching problem ([33]), which is a two-layer structure and built on the simple series synopsis, the mean values of the disjoint windows. Using disjoint windows is faster to update than sliding ones. The constraint normalized subset matching problem provides a knob to control the offset shifting and amplitude scaling flexibly. When storing event data in existing Time-Series Databases (TSDBs), the retrieval may have performance problems. Also, existing TSDBs do not have a specific query language defined for event analysis. [34] develops a model for event specifications and use it to specify abnormal system states to be captured to allow timely mitigation. The event model is integrated into the TSDB by translating them to continuous queries defined in some TSDBs. The authors of [34] develop a model of event specification since TSDBs do not have specific query language. In [35], the authors study the approximate range emptiness problem, which answers an emptiness query of the form "$S \cap [a;b] = 0$?" for an interval [a;b] of length L $(a, b \in U)$, over sliding windows in the IoT data streams. [36] designs a real-time and historical multi-view IoT trend system. Using an information graph, it displays the relationship between different sensor data and, using the method combining real-time and history, it proposes a buffered batch data loading algorithm to form a dual-display mode system. About data quality validation, the contribution of [37] is to enhance the stream processing system such as C-SPARQL with production rules, instead of statistical approaches, to achieve a Continuous Time-Aware reasoning. It is used to provide stream validation for the quality problem relating to inconsistencies in sensor streams. Errors in measurements may cause bad procedure (and even sometimes dangerous) triggers.

Spatio(-temporal) retrieval

[38] aims at designing a distributed framework of massive trajectory data analysis based on Hbase to realize spatio-temporal query more efficiently for urban computing scenarios. They design a temporal-based pre-partitioning strategy to improve the performance of data written. Then they develop a Multi-Level Index to speed up the process of spatio-temporal query. [39] consider the case of IoT databases along roads to organize the storage according to road segment, instead of the cell, as the unit of space mapping and data storage so that data requested in a road query can be stored together to enable efficient I/O. Then, one can efficiently found the units covered in a road query, which is usually concerned only about data on a few segments of roads. [40] offers location-aware services based on data messaging and aggregation techniques. Its authors design a taxonomy for the classification of the IoT devices based on their mobility frequency and leverage it to design a priority scheme to address the co-located devices that offer similar services. In [41], considering that stream processing must not be limited to the temporal dimension but also consider the spatial one, a query language is developed to process temporal and geographical data, based on an index taking into account both dimensions. With in-network processing, generated queries are routed to the nodes belonging to the specific region of interest, mainly using standard WSN (Wireless Sensor Network) geographic routing algorithms. The IoT IETF standard routing algorithm RPL (Routing Protocol for Low-Power and Lossy Networks) builds a tree-like routing topology that is inefficient for a spatial type of query because it floods the message to all nodes either inside or outside the target region. The authors of [42] improve RPL in particular by considering spatial partitioning. IoT applications use more or less two kinds of data: either time-series or contextual information (or graph), mainly stored through

massive data (times-series) or graphs and ontologies (contextual information). [43], [44] and [45] address the issue of designing search engines including both spatial and temporal dimensions, for example, using dual temporal and spatial stores.

2.2.3 IoT architectures, data and query distribution and caching

It is critical to perform communication-efficient data aggregation to answer complex queries (e.g., skyline queries and equality joins) from IoT applications. [46] investigates the problem of constructing an aggregation tree for complex queries with the minimum communication cost. [47] addresses the issue of data replication and keeping only necessary data to process them and respond in a limited time. Nodes can act as a team and cooperate to store the data close to processing tasks defined as queries. Every node decides, in real-time, if the observed data are correlated with the available datasets or if they are outliers. When data are accepted to be locally stored, nodes select their peers where data will be replicated. [48] proposes a multi-attribute aggregation query mechanism in the context of edge computing. [49] considers the problem of query placement on edge and cloud resources for dynamically arriving and departing analytic dataflows. This is an optimization problem to minimize the total makespan for all event analytics while meeting energy and compute constraints of the resources. [50] proposes a framework aiming at reducing the energy at the sensor level and the billing cost at the cloud level. [51] designs a centrally managed distributed infrastructure based on the state of the art big data technologies for processing many real-time queries. [52] and [53] design multitenancy cloud for IP traffic flow. [54] addresses the issue of time-series database in the edge between the sensors and the cloud, taking into account the low space and processing capabilities of the edge. [55] employs the parallelism in edge computing environments to facilitate the top-k dominating query process over multiple uncertain IoT data streams. The challenges of this problem include how to quickly update the result for processing uncertainty, reduce the computation cost, and provide highly accurate results. In [56], an edge-centric architecture is considered, and a routing method is proposed to find the information efficiently. [57] proposes a multicast algorithm that is data-centric (based on feature information) and intended for low power networks. [58] and [59] propose semantic-based algorithms for routing and IoT service discovery (with appropriate information compression). In order to save bandwidth, [60] uses a gossip-based protocol for nodes to organize into groups based on attributes and current value. [61] addresses the issue of resource allocation in heterogeneous edge/fog devices. [62] takes advantage of data aggregation and both spatial and temporal reuse by exploiting long-term static scheduling for periodic data to ensure the latency and data rate and by employing short-term dynamic scheduling for event-driven, query-based data to improve transmission efficiency.

[63] implements caching in cloud for frequent n-hop neighbor activity regions.[64] uses proactive caching in placeholders to decrease query delays. According to [65], the data queries often are similar or overlapped to the previous queries, especially when the user adjusts the query parameters on the same time-series data for displaying on a visualization tool. The article proposes a cache mechanism to help reduce the query response time if the current query range overlaps with the previous query ranges. [66] observes that memory object caching are inefficient in case of unrepeated queries, and it proposes pro-active caching. The prefetching is done according to rules customized according to the workload.

2.2.4 Interoperability

The authors of [67] describe a semantic data model, rules, and a reasoning platform taking SPARQL queries as rules to enable high-level data abstraction and knowledge extraction. [68] integrates existing ontologies to provide semantic interoperability among heterogeneous devices and users in the healthcare domain. [69] focuses on facilitating the understanding of messages exchanges between artefacts founded in different ontologies by semantic translation based on ontology alignment. In [70], the authors leverage semantic technologies like Linked Data to disseminate data to relevant data consumers. Time-series databases are used to store IoT data, but they lack a good semantic model to support data sharing. That is why [34] proposes a monitoring data annotation model to guide the systematic specification of monitoring data streams. In [71], the authors propose IoT-Directory, which makes it possible to manage packet brokers using different communication technologies and formats in an ontology-based way, and in particular, to ingest data from them in the IoT-Directory. Another issue is the necessity for processing IoT data from multiple heterogeneous data stores: in [72], a multi-store query system for IoT data is proposed. The IoT is characterized by a wide variety of data sources, large scale and heterogeneous structures. However, those characteristics bring great difficulties to the storage and rapid retrieval of IoT data. By considering the common attributes of IoT data, based on plug-in ideas, combined with Redis and HBase, the paper [73] proposes a framework named HSFRH-IoT, which solves the problem of efficient storage and retrieval of massive heterogeneous IoT. [74] presents an ontological model which improves semantic searching and querying capabilities by hiding the heterogeneity of entities and their produced data. [75] addresses the issue of semantic relationships between the data over data/event streams using Complex Event Processing to derive time-annotated RDF data from the basic events streams and C-SPARQL for processing online queries over a domain ontology for property inference. In [76] an agent-based server system approach, which improves the resource sharing between heterogeneous WSNs in IoT/CPS (Cyber Physical Systems) providers, is proposed. [77] presents a versatile architecture of a broker named X2CBBR and based on XML-base publication data and Xpath-based subscription data that can operate in IoT with different publish/subscribe systems. The EPC network is a collection of industrial standards designed to build an Internet of physical objects. ONS, a directory based on the DNS, is one of the important components of the EPC network. ONS provides a connection between the product code and Information Services in IoT systems. [78] proposes an extension of ONS architecture to support heterogeneous object code identification dynamically.

2.2.5 Data contextualisation and discovery

Contextualisation

A classical approach is to use the data context to facilitate the query: [79]. Contextualization of IoT data is the technique that excludes irrelevant IoT data from processing and dissemination [80]. It involves query rewriting, semantic web and rule-based context management or ML and data science-based solutions. [81] proposes a generic approach to represent and query situations in IoT environments. [82] performs the reasoning process to change existing data on the ontology according to the rule created and produce more representative and contextual data. [83]

present an application that allows users to look back on their memories by prioritizing photographs that were taken in contexts that are similar to the current context. It is built on a contextual database middleware that provides context storage and context-sensitive queries that determine the similarity between pairs of contexts. Reminisce uses the contextual database to store the contexts of a user's photos, including location, time, neighboring devices, and weather conditions. Later, the app identifies the photos whose stored contexts are most similar to the user's current context by querying the contextual database.

Context sharing and discovery

Applying semantic techniques to IoT can support interoperability, collection, annotation, validation, effective data access, integration, resource discovery, but also processing and storing as well as reasoning and querying for data knowledge extraction ([84],[85]). The IoT requires a semantically rich, dynamically extensible, low complexity service discovery mechanism. It differs from classic discovery approaches such as UPnP or DNS-based approaches which are well suitable for connecting resource constraint devices but do not match the required semantic richness of the IoT. [86] present a solution that is extensible at runtime, contrary to mechanisms such as UDDI (Universal Description Discovery and Integration). It is based on a distributed modular directory of service properties. [87] proposes a platform enabling smart things and IoT silos to discover, validate and share relevant and dependable context. [88] proposes a distributed service discovery algorithm intended for a highly dynamic network environment. In [89], a distributed model for resource discovery in IoT is based on a structured peer-to-peer scheme, which supports multi-attribute queries and uses a distributed hash table as an overlay to organize the discovery process. In [90], ontologies like W3C's (World Wide Web Consortium) and OGC's (Open Geospatial Consortium) Semantic Sensor Network (SSN) and Sensor, Observation, Sampler, and Actuator (SOSA) are extended to annotate IoT streams for search and discovery. To share and publish the domain knowledge of IoT objects, developing a semantic IoT model-based directory system that manages meta-data and relationships of IoT objects is required. [91] proposes a directory that supports semantic description, discovery and integration of IoT objects. [92] proposes semantic accessors, including a local semantic repository for maintaining the context, accessors for querying and dynamically updating the repository and servicing. [93] presents an approach covering both service discovery and invocation for IoT-aware processes. It proposes a novel concept for integrating semantic queries into process activities to support runtime discovery and dynamic invocation of IoT services. It extends a generic process meta-model by a set of process step types supporting SPARQL queries. This approach provides runtime flexibility in terms of resource allocation and therefore increases the reusability of process models. With this solution, it becomes possible to specify commands like "switch on all lights in the kitchen" directly in a process step without knowing which Things are capable of executing this command. In [94], the authors speed up the discovery process by centralizing the RDF semantic information instead of keeping it distributed in the resource tree.

2.2.6 Query performance

In [95], the authors propose an index to compare query languages from both performance and expressive points of view, taking into account the « richness » of a

query. In [96], IoT-Lite proposes a minimal lightweight semantic. In [97], a semantic query engine for Industrial IoT, SqenIoT, is designed together with multi-level ontologies and mechanisms and a things query language targeted for resource-constrained gateways and non-technical personnel. In [98], IoTm, a framework for measuring IoT performance metrics, is presented, which include both IoT network's Quality of Service (QoS) metrics and IoT node's resource utilization metrics. IoTm has two key properties: 1) it is lightweight and thus amenable for implementation on resource-constrained IoT nodes; 2) it can perform measurements at fine-grained levels and not just at aggregate levels. [99] an advanced high-level formalization - Performance Evaluation Process Algebra (PEPA), a kind of stochastic process algebra (SPA), is adopted to model the process of real-time traffic status query in Intelligent Traffic System (ITS).

Meanwhile, using the fluid approximation approach to analyze the performance of the model, and then the performance parameters of the system in practical applications can be achieved accurately. A dedicated testing procedure has been configured in [100] for evaluating InfluxDB, one of the most effective and widespread TSDBs. The performance analysis, carried out on a specific use case, demonstrated that the database write and read performance can be significantly affected by the used data model, with queries executed on the same data requiring times from hundreds of ms to seconds in the worst cases.

2.2.7 Methods

Probabilistic approaches

[101] address the issue of aggregation query (e.g. give the maximum pollution level in an area) via probabilistic sampling. Reverse Nearest Neighbors query finds the objects that have the query as the nearest object and plays an important in many applications. [102] thus studies probabilistic Reverse Nearest Neighbors query over the uncertain data streams. IoT is used to monitor natural phenomena, but spatial queries are costly. Efficient sampling-based approaches are proposed in [103]. To balance the retrieval efficiency and the maintenance cost, [104] adapts the probability that a thing property is indexed to its change frequency and also to its query frequency. [105] uses a probabilistic flood search algorithm to find devices distributed in heterogeneous and dynamic environments. Sampling-based approximate data analytics are the most widely used, which trade the output quality for efficiency. [106] proposes a real-time data flow system for edge computing and analysis.

Fuzzy approaches

[107] uses fuzzy ontology query and reasoning to generate complex event query plans, and context-aware queries are rewritten into context-independent sub-queries. Data windows are partitioned according to different event patterns and contexts. The sub-queries are optimized and executed in parallel based on data partitioning.

Machine-learning methods

[108] consider a neuronal approach to minimize storing and processing of image data in IoT using neural networks to extract relevant information. To achieve efficient storage management, [109] classifies and reduces data using a Support Vector

Machine (SVM) classification algorithm.

Graph methods

In recent years, the Social IoT paradigm, where objects establish social-like relationships, has become popular as it presents several interesting features to improve network navigability and implement efficient discovery methods ([110], [111]). Using social relationships can help to accumulate experience knowledge and also optimize the network traffic load ([112]). In [113] various graph algorithms are used to respond to the user's queries smartly. In [114], graph databases in clouds are used to represent data. [115] addresses the problem of community search in social IoT, which is beneficial to resource/service discovery, from the viewpoint of a dense subgraph query. In [116], the issue of faster query processing of data is addressed. It presents a Log Based Method (LBM) to store and query IoT data in Resource Description Framework RDF format. LBM exploits skewness in access patterns of records by analyzing query workload and partitions the basic triple table into hot and cold sections. To facilitate data retrieval and analytic, [117] ranks the web pages where the information is stored in function of the quality of the content and the interconnection between the web pages.

Blockchain

[118] designs a distributed IoT blockchain ledger for managing the metadata needed to describe IoT devices and the data they produce. It is not controlled by any individual or organization. The paper also proposes a marketplace that provides the functionality needed for IoT device registration, query, integration, payment and security via the proposed blockchain. [119] proposes a solution called Semantic Smart Contracts (SSC), which integrates RESTful semantic web technologies in smart contracts, deployed on the Ethereum Blockchain platform, for indexing, browsing and annotating smart contracts. The solution also exposes the relevant distributed ledgers as Linked Data for enhancing the discovery capability.

2.3 Scalability

IoT faces well-known scalability issues. EDs hardware have small power resource, limited memories and processing capabilities, bandwidth is either sparse or it may be overloaded due to the huge quantity of transmitted data. Legal regulations impose limitations on the spectrum usage (e.g. respect of duty cycle constraints). Most of the problems arise at the network front-end but also in the back end since all the data are finally routed to servers or clouds on which they are stored and processed ([120], [121]). Thus, economising bandwidth, batteries, memory, storage and processing are ways to increase IoT scalability. Compression techniques are solution but also subtle tradeoff between information loss, energy, storage, time and bandwidth minimisation ([122]). Most of the time these efforts lead to design new compression methods ([123], [124], ([125], [126], [127], [128], [129], [130], [131], [132], [133], [134],[135], [136], [137], [138], [139], [140], [141], [142], [143], [144], [145], [146], [147], [148]). Some are lossy while others are lossless. Deleting redundancy, data aggregation, quantization, approximation by simpler functions, use of dictionary-based methods, entropy coding, transform methods, compressed sensing and neural network approaches have been applied for IoT.

The easiest way to reduce data transmission is to delete redundancy or to round them to near values. Sensors often generate time-correlated data. For example, the temperature may vary slowly. Run-Length Encoding (RLE) takes advantage of the adjacent clustering of symbols that occur in succession. It replaces a "run" of symbols with a tuple that contains the symbol and the number of times it is repeated. The authors of [142] apply Delta Encoding followed by RLE at the end node. At the gateway tier towards the back end, they apply the hierarchical clustering for grouping datasets received from sensor nodes dependent on the Minimum Description Length (MDL) principle. If any pairs of received datasets can be compressed by the MDL principle, they will be combined into one cluster. For every cluster, it is preferred to look for a model or representation of the dataset to compress the others based on it. They call this model or representation a hypothesis. For studied each cluster, they send the hypothesis followed by the difference vector between the hypothesis and the other datasets belonging to this cluster. In [135] delta compression allows sending only the difference between two consecutive temperature measurements. In [146], only significant differences between surveillance video frames are sent. In [126], GPS locations are shortened by substracting the common latitude and longitude parts of the measured positions. Data are also usually quantized to round the measures to significant approximations requiring fewer bits for coding. They are often aggregated ([126] or [129]). Also, fields of measured IoT data may change at different speeds depending on the nature of the measured quantities. Observing the change in frequency allows adapting the way they are sent or pruned ([132]). The paper [136] uses aggregate approximation based on Symbolic Aggregate approXimation (SAX) ([149]) and symbol encoding. The idea is to replace fixed sized data windows by their average, and successive averages are differentially encoded and replaced by codewords. Also, in [130], the authors suggest that applying repeated pattern-based graph compression techniques may be used to compress IoT data represented by graphs.

Approximating the measurements reduces their size also ([127]). One can compress signals by approximating them with auxiliary, more straightforward functions. Lightweight Temporal Compression (LTC) ([150]) is an energy-efficient lossy compression algorithm that maintains memory usage and per-sample computational cost in O(1). LTC estimates data points using a piece-wise linear function that guarantees an upper bound on the maximum absolute error between the reconstructed signal and the original one while maintaining memory usage and per-sample latency in O(1). Such an LTC method is proposed in [143] which increases the compression efficiency by exploiting the latitude offered by the error bound to find a better approximation in terms of compression ratio. Fan algorithm ([151]) is an adaptive sub-sampling approach that operates by drawing the longest possible straight line between the starting sample and the ending sample in such a way that the error in reconstruction of the intermediate samples are less than the maximum specified error value. In [137], the authors take profit of the high compression ratios of lossy compression with lossless entropy coding compression for the resulting error compressions. For the lossy compression, any lossy algorithm could work, but it must be implementation efficient (in terms of energy, memory...). The authors choose the Fan algorithm. They use Huffman coding for the lossless compression algorithm.

Classical compression approaches based on dictionaries or entropy coding have been adapted to IoT. In [145], a specific dictionary is created for different kinds of data depending on their change frequency. In [124], a dictionary is created online at run time for biomedical signals. In [138], the LZW compression algorithm is applied

to the differences between sample measures. In [131], an offline frequency distribution is used to create a symbol-code lookup table. They use an extensive set of data from a previous study, and they present an analysis of the entropy of activities of daily living accelerometer data. Lossless Entropy Compressor, a Huffman based compression method, has been made dynamic to handle possible heterogeneity and changes of the signal ([139]). Obviously, depending on the operating conditions, the best among uncompressed, lossy or lossless modes can be chosen dynamically ([123]).

Transform methods are classical tools for compressing data but might be CPU resource consuming. [120] evaluates several lossy compression algorithms for efficiently storing weather sensor data based on the encoding of temporal changes and three signal transformation algorithms on spatial data. Specifically, they evaluate the fidelity of reconstructed weather sensor data using Discrete Cosine Transform, Fast Walsh-Hadamard Transform and Discrete Wavelet Transform (and also Lossy Delta Encoding). The objective is to provide useful information for minimizing data reconstruction errors, and more importantly, make sure they are within a tolerable range. Chebyshev compression is considered in [121] and [141].

Compressed sensing is a new technique. As stated in [152], in a series of pioneering works by Candes ([153], [154],[155]), and their co-authors, it was shown that when a signal has a sparse representation in a known basis, one can vastly reduce the number of samples that are required—below the Nyquist rate and still be able to recover the signal perfectly (under appropriate conditions). This framework suggests compressing the data while sensing it; hence the name compressed sensing. Nevertheless, on the one hand, compressed sensing reduces the number of measurements and the sampling rate, but on the other hand, it increases the computational complexity of the signal recovery ([156]). Actually, the signal is approximately recovered by solving a convex relaxation of a non-convex optimization problem. [134] proposes a unified approach for compression and authentication of smart meter reading in advanced metering infrastructure. In [133] an algorithm is designed which combines the accuracy of standard lossless compression with the efficiency of a compressive sensing framework. It balances the tradeoff of each technique and optimally selects the best compression mode by minimizing reconstruction errors, given the sensor node battery state.

Recently, Neural network-based techniques entered the landscape of IoT data compression techniques. In [144], data are compressed by their regression curve obtained from a neural network. In [157], biomedical signals are compressed using autoencoders. These neural networks are three-stage networks whose input and output dimensions are the same, while the hidden stage has a smaller dimension. The output of the first stage has thus a reduced dimension compared to the input and constitutes the compressed data. Prediction methods are also used. Actually, neural networks are known as universal function approximators with the capability to learn arbitrarily complex mappings, and in practice, show excellent performance in prediction tasks. Thus, the authors of [147] train a Recurrent Neural Network (RNN) predictor followed by encoding with a traditional arithmetic coder block using the probabilities generated by the trained neural network. The decompression is performed symmetrically and requires the trained model for arithmetic decoding. In [158] a prediction scheme is implemented on cluster nodes and cluster heads to reduce data transmission. If the measured data corresponds to the predicted one, it

has not to be transmitted. Neural networks (NNs) and, more specifically, LSTMs are proposed to perform predictions.

ML can simplify signal detection by training a general data-driven signal detection model. However, fully connected neural networks would introduce processing latency and extra power consumption, making them unsuitable for deployment on IoT devices. That is why neural networks themselves must be compressed. Therefore, the motivation of [159] is to investigate different neural network compression schemes for system simplification. Three compression strategies are studied, including topology compression, weight compression and quantization compression. These methods show efficient neural network compression with tradeoffs between computational complexity and bit error rate (BER) performance. Also, compression technology through pruning research has gained momentum and is an important tool for improving performance during inference. [140] focuses on pruning unwanted filters and nodes in all layers of a network. The network is pruned iteratively during training, and a significant number of filters/nodes are removed while ensuring any loss in accuracy is within a predetermined range. Other methods exist like efficient convnets, ThiNet and Cross-Entropy Pruning.

Most of the time, IoT solutions are scenario-specific and application-specific and rely on standard layer-2 solutions like LoRaWAN or IEEE 802.15.4. One example of such application-specific solution is [148] that proposes improvements to LoRaWAN communications by fine-tuning the communications parameters to the device's situation (location, number of RGs, scheduling...). However, for interoperability purposes, IPv6 is an obliged interconnection solution, and it raises the issue of header overload. Header compression algorithms like Robust Header Compression version 2 (ROHCv2) from Request For Comments (RFC) 5225 may solve this problem. It sends context information packets like "initialization and refresh" to transmit persistent information to the receiver and consecutive compressed "sequential" packets, which can be decoded from the previously transmitted context. Nevertheless, if context packets are lost due to transmission errors or overflows, sequential packets cannot be decoded. Also, most classical compression algorithms (e.g. delta compression, LSB compression, table-based compression...) assume that an up-to-date reference value exists on the decompressor side; thus, they face the same problem in case of possible packets losses. The solution the authors of [160] is to prepend context information onto sequential packets using Random Linear Network Coding. Another solution is SCHC [3] which is a framework that provides both compression and fragmentation functionalities. It is being standardized by the lpwan [161] working group at the IETF. RFC 8724 and its related works [161] at the IETF are also closely followed by other SDOs such as IEEE 802.15, LoRa Alliance. It is considered an efficient solution to connecting the LPWANs using IPv6, thus enabling end-to-end IP connectivity. With the help of the SCHC framework, it is possible to compress an IPv6 header from its original size of sixty bytes down to two bytes, thus reducing bandwidth usage and increasing communication efficiency. In [162], SCHC is used as a unifying transport layer for multimodal LPWAN solutions. The main drawback of SCHC is that it works under the assumption that LPWANs are preprogrammed with known data flows, thus uses a static context. However, this assumption is not always valid in an IoT context, where devices should be accessible from any IPv6 address. In [163] a dummy mapping technique that can effectively compress/decompress some header fields of unknown flows is proposed. The dummy mapping creates a fixed-size list of dummy values mapped dynamically at the gateway compressor to the values of headers fields.

Also, reducing traffic by compression contributes to scalability and optimizing signalling and header information transmission. Over dissemination of packets during Neighbor Discovery (ND) and headers with larger sizes lead to prodigal power consumption leading to short network life in 6LoWPAN (cf. [164]).

2.4 Security

2.4.1 Generic Approach

Security of IoT devices and infrastructures is an extensively studied subject; there are articles and surveys regarding many aspects of security in IoT, taking into account the strengths, weaknesses and specificities of IoT devices to point out new research opportunities and concerns to improve security.

[165] highlights various needs regarding security for IoT and points out the need to build tailor-made security measures adapted to devices resources, embedded protocols and system specifications. An embedded security framework is also provided to propose a co-design methodology that helps design security features by considering their hardware and software constraints. [166] responds to a need for standardization and documentation regarding general points of vigilance when building IoT solutions. It provides a state of the art regarding security threats and risks for IoT as well as security guidelines.

[167] studies the vulnerability introduced by bridges between the Internet and non-IP devices connected to a HN. By introducing new hardware or protocols into the network, the system creates attack and defense scenarios that need to be extensively studied as devices need to handle attacks from more powerful attackers. [168] uses a similar approach, compares security threats in the IoT, discuss IoT scenarios, analyses possible attacks, and provides insight on security threats and vulnerabilities in the IoT environments. One of the most documented in such IoT environment are sinkhole attacks[169]; these attacks rely on the trust built among devices in a wireless network to build their routing strategy. [167] already evokes them, but an extensive study of sinkhole attacks and how to defend against them is provided by [170]. [170] documents how to detect and mitigate these attacks, how to balance security and usability for devices and which tradeoffs are necessary for these scenarios.

[171] and [172] document LoRa and LoRaWAN security vulnerabilities and various attack scenarios. [171] focuses on the LoRa technology, studies the LoRa network stack and provides attack scenarios built on existing hardware and technology. [172] tackles LoRaWAN and points out its vulnerability to replay, man-in-the-middle and battery exhaustion attacks by analyzing the packet exchange between LoRaWAN devices and backend. The attacks scenarios mentioned are put into test through various proof-of-concept.

Rather than focusing on attack scenarios or building security countermeasures specific to given hardware, other papers focus on generic security approaches built regardless of the underlying technology. [173] proposes an All-IP IoT architecture to support a complete IP adaptation method for IoT applications around four key topics: mobility, web enablement, time synchronization, and security. [174] indicates that IPv6 will become the de-facto standard for interoperability in IoT and point out security challenges related to processing power in the context of lightweight cryptographic algorithms and standardization. The paper studies the complete IP stack at large regarding IoT constraints and provides an overview of possible solutions to

improve IoT systems at each layer. [174] also evokes issues regarding authorization and privacy concerns which will be detailed furtherly.

2.4.2 Authentication

[175] proposes a key management solution to handle the IPsec key exchange as well as support 802.15.4 channel securitization.

[176] details specific needs for medical IoT regarding authorization, privacy requirements or data confidentiality, taking into account specific constraints from resource-constrained devices. The paper proposes a context-aware security architecture handling access control and privacy-preserving data silos and provides an access control engine running on resource-constrained sensor nodes.

[177] details key exchange scenarios in WSN to establish a secure channel to protect the data transmitted over the air. Various scenarios are described, such as public-key cryptography and PSK. Backend considerations are also considered. [178] studies identity management systems to support data integrity protection and proposes a framework for authentication suited for IoT environments.

[179] propose an authentication mechanism based that uses multicast communications to save transmission during the authentication handshake for resource-constrained devices, evaluating its performances and analyzing the security paradigms in place to support IoT development despite its constraints. [180] is a reflection on authentication, authorization and accounting on constrained devices by leveraging the PANA (Protocol for carrying Authentication for Network Access) paradigm, which was ongoing work at IETF back in 2013. The paper provides a study of the EAP/PANA paradigm in constraint devices and an implementation that serves as a proof of concept to both the IETF standardization community and the scientific community. The paper is based on a Contiki implementation of EAP/PANA called PANATIKI based on simulations and results from an experimental testbed.

[181] presents a two-way authentication scheme based on the Datagram Transport Layer Security (DTLS) protocol. The authors provide a solution based on existing standards and protocol to embark the algorithm on 6LoWPAN EDs and evaluate the solution through experiments. [182] targets V2X communications building a delay-aware, reliable, scalable and secure network. The paper aims to provide insight on the handshake mechanism to build for such systems considering their constraints (mobile devices, protocol overhead, transmission delays) while keeping the system reliable. The proposed mechanism reduces delays in the handshake mechanism by leveraging the capabilities of edge infrastructure to reduce delays and improve the efficiency and reliability of vehicular communications.

[183] uses the CoAP protocol to propose an authorization framework that can build authorization based on an external OAuth authorization service (OAS). The proposed scenario benefits from the OAS's strength while keeping the solution with IoT's usual constraints: low processing load, customizable and easily scalable. The solution is not energy efficient due to the multiple transmission necessary to fragment the messages but is efficient regarding other primordial aspects such as memory footprint. The solution is also efficient for developers as OAuth is a well known, well-documented technology and the solution benefits from the existing implementation of the OAS paradigm.

Scalable Authentication and Authorization framework compatible with different IoT technologies :

Authentication, Authorization and Accounting (AAA) architectures are usually consolidated in a single centralized database [184]. The centralized AAA framework has its advantages but also has significant disadvantages, such as creating a single point of failure, and the issue of consolidating all user information into one database would probably go against European's General Data Protection Regulation. Blockchain using distributed ledger has been experimented and deployed[185][186] to accomplish a scalable de-centralized Authentication and Authorization AAA framework. However, the blockchain model has several drawbacks as a feasible operational model [187] in an open/global scenario.

eduroam uses the distributed Public Key Infrastructure (PKI) based on X.509 digital certificates for AA. The PKI model has been tested both on the Internet and in IoT for dynamicity and scalability. The primary issue with the X.509 digital certificates is their size and compatibility with resource-constrained IoT networks. Since our focus is on Organizational Interoperability between the networks in the federation operating in the IP space, we plan to employ PKI for AA.

eduroam uses a combination of IEEE 802.1X, the Extensible Authentication Protocol (EAP) and RADIUS to provide [188] AA of the Wi-Fi ED to the network. The trust fabric in eduroam is a PSK between the RADIUS servers (organizational, national, global) based on the DNS hierarchy. The organizational RADIUS servers agree on a shared secret with a national server, which agrees on a shared secret with the root (i.e. global) server. The RADIUS hierarchy forwards user credentials securely to the users' home institutions for verification and validation.

Such a trust fabric, wherein there is a PSK to be shared hierarchically, hinders the federated model that we envision for IoTRoam. Different IoT networks use different mechanisms to share the PSK between the ED and the AA servers on the Internet to securely onboard the ED in the HN. Forcing them to transition to a newly proposed PSK mechanism is not operationally possible since multiple stakeholders are involved. For secure onboarding of the ED, we will use the existing mechanisms used by the respective IoT technologies and propose a global mechanism tested on the Internet to mutually authenticate different servers (in IP space) in the federation involved in ED onboarding.

Using a combination of an existing PSK mechanism and a global PKI mechanism proposes a design that satisfies the requirements outlined by WBA for a scalable (even when there are millions of networks interconnected in the federation) AA framework applicable for all IoT technologies/applications.

Authentication at IETF :

IETF worked on designing authentication frameworks, one of them was the Protocol for Carrying Authentication for Network Access (PANA) [189] designed by the pana working group. According to its charter[190], the pana working group aims to build generic frameworks to support authentication in various scenarios around mobile IP networks. The working group proposed 6 Standard RFCs detailing each component necessary to build such a security framework as part of its work.

Authentication frameworks were also proposed for adoption to other working groups, [191] proposed to the Constrained RESTful Environments (core) working

group, then to the Authentication and Authorization for Constrained Environments (ace) working group to work on Delegated CoAP Authentication and Authorization Framework (DCAF). The objective of DCAF was to delegate authorization and authentication information to a trusted third party with less processing limitation than the constrained device that requires it.

2.4.3 Trust

Trusting a third party or a communication peer is an important issue when discussing security in general and IoT security in particular. IoT's constraints regarding processing power and needs for energy efficiency usually come with a necessity to rely on others to delegate certain operations. Building and evaluating trust frameworks when working with constrained devices has been a subject of interest since the beginning of IoT research.

[192] summarizes the issues and requirements for building trust within heterogeneous networks. The main topic studied is building trust regarding routing protocols and trusting peers when bridging the connection. Another studied aspect of building trust to decide where to route the traffic consists in creating a network trust model to decide which level of trust a device gives to another peer. In trust delegation models, another critical aspect is deciding on a key exchange and management policy; the paper evokes possible protocols to exchange keys and negotiate trusted channels.

[193] proposes a trust management model based on building a fuzzy reputation system within the IoT network. The article extensively studies trust in networks, then uses the NS-3 simulator to generate traffic and build trust within the IoT network. The proposed mechanism, TRM-IoT, performs better regarding routing performances, packet loss and other network metrics while keeping the system lightweight and within IoT specific constraints but seems less reliable when attacked by malicious third parties. [194] proposes to establish trust models by building social reputations between devices the same way reputation is built within human social networks. Contrary to [193], the simulations from [194] show many benefits regarding excluding malicious nodes within the network based on a feedback system that evaluates trust level based on nodes centrality.

[195] studies WSNs in a countryside environment, builds a trust framework and experiments upon it. The model shows interesting results regarding data reliability and isolation of malicious nodes.

[196] identifies trust issues within heterogeneous network environments and builds a trust consensus based on social IoT paradigms in a similar way to [194] but focusing on other key aspects on building trust within a network such as handling large scale infrastructure and taking into account devices capabilities in terms of storage and processing power. The approach is studied furtherly in [197] with additional analysis regarding trust assessment and trust convergence time. The tradeoff between these two is studied and show significant results compared to non-trust-optimized systems.

[198] concerns itself with building reliable IoT infrastructure while keeping the system privacy-preserving and building trust into the network. Their system relies on a trust policy handler that defines operations between devices, a key management component that assists in securing communications, an identity manager and

a reputation manager that assists in trust management within the network. Their system provides a generic approach to IoT security but, as they point out, lacks an actual proof-of-concept. [199] described extensively the issues and solutions regarding trust management in IoT and [200] surveys trust establishment in IoT.

2.4.4 Privacy

[201] proposes a framework to measure possible invasion of privacy in protocols using a mathematical method. However, this method does not apply to all protocol as the protocol need to fit certain algebraic properties but should apply to IoT specific solutions. [202] proposes a technical solution to enforce privacy protection using a middleware system as a privacy broker. [203] sums up many challenges within IoT systems regarding privacy protection and security. [204] aims to improve privacy protection in IoT using a solution that gives control to the user and then designing complex privacy policies and filters. Such a solution aims to identify finely all the elements that need access to personal data and which specific access they need.

The question of privacy is not only a technical issue to monitor but a transverse issue to address from various experts, not only from the computer science field. [205] is a multi-disciplinary approach study that reflects on privacy and security of IoT hardware, software and protocols from a regulation point of view. It extensively studies sensors networks and the possibility of break privacy protections, creating discrimination, or making more secure devices with a combined juridic and technical approach.

[206] proposes a novel approach around Quantum Lifecycle Management (QLM) that allows devices to keep communicating, bypassing firewalls. Such a solution, should it work, would prove interesting regarding access to information from isolated devices but also pose real threats regarding security from a generic point of view, especially with regards to the issues discussed by [205].

To support all threats identified in the previous papers, [207] proposes a conditional privacy-preserving authentication with access linkability (CPAL) for roaming service. The idea behind CPAL is to offer a secure roaming service that respects user's privacy using a multi-layer approach to information access within the system. The system and its performances are also studied to prevent the solution from being too heavy for IoT solutions. [208] also defines a roaming protocol for IoT solutions exploiting the strength of edge computing to authenticate the devices. The solution seems to protect against various known attacks, and simulations underline interesting authentication delays and energy consumption results.

2.5 Coverage and Roaming

Device mobility management is one of the most important points when designing an IoT system. When working with things as fridges or doors, mobility would not be significant. However, mobile devices such as sensors, watches, phones, luggage or cars should be provided coverage as often as possible.

Many parameters are considered when handling mobility, such as covered devices' output power, antenna direction, data payload, speed, size, environment, and duty cycle. Much of these parameters are also encountered on the RG side.

Coverage evolution is a well documented subject (cf. [209] [210] [211] [212] [213] [214] [215], [216]). As IoT developed outside of its small scale industrial use, the need for coverage became prominent and new technologies were introduced [217] to increase the scale from which to connect things [218]. Consumer IR, Bluetooth, Wi-Fi, which are still developed as of today. But also long coverage communication ([211] [213]) ranging up to tens of kilometers such as LoRaWAN (cf. [219] [220]), Sigfox, Narrow-Band IoT (NB-IoT). Network topology may also change the way mobility is handled. [209] gives a unique perspective on the way to improve coverage and mobility, though Machine-to-Machine Technologies, listing the then-emerging Unlicensed LPWAN and evolutions to the 3GPP Standard (also known as LTE-M and NB-IoT). Vehicular networks are an excellent example of the evolution of M2M networks and how mobile M2M networks can use their environment's infrastructure as good support to mobility [221]. [210] after an overview on various IoT technologies, from short-range technologies such as ZigBee, Bluetooth and Wi-Fi to long-range such as GSM, LTE and Satellite, provides a complete overview of LPWANs offers in the context of Industrial IoT and listing their strength and limitations. LPWANs are presented as a good opportunity regarding coverage, with an evolving community developing interesting standards and with increasing deployment. LPWANs also enable new channels to communicate as a substitute or a complement to the existing GSM networks [211]. Coverage was also studied through simulations, [222] provides an extensive study on LoRaWAN performances through a study on packet delivery in a realistic context and aim to provide reference assumptions and numbers for people willing to modelize a LoRaWAN infrastructure. [216] expands this work by providing numerous measurements on link quality based on actual LoRaWAN payloads, studying Packet Reception Ratio based on Packet Length, Spreading Factor (SF) or Signal to Noise Ratio. Their experiments show that the most significant impact on LoRaWAN communication comes with the scalability of the solution as the recommendation is to group data transmit long payload instead of short chirps to prevent packet losses from interferences from other devices. [212] studies theoretical LoRaWAN and NB-IoT coverage for connecting smart meters. The solution aims to cover a whole population with various population densities (urban, suburban, rural) and device positioning (deep indoor, indoor and outdoor). This document also points out the importance of network deployment to enable an efficient service.

On the other hand, development on roaming was less significant compared to coverage. Roaming is an essential factor if one wants to improve its mobility. While coverage is important when considering small scale mobility (neighborhood, warehouses...), roaming is the technology to develop to improve coverage for things moving far from their HN. Roaming allows building a network of interconnected antennas from various partners to provide universal access to the network.

This lack of development regarding roaming in IoT-enabling technologies is pointed out by [209] as roaming seems to be an excellent way to improve coverage in Smart Cities. This is especially important in a dense area where antennas might interfere a lot.[210] while providing many details regarding coverage, data rates and energy consumption, does not provide much input regarding roaming. Only pointing out that each technology handles roaming. [223] draws a comprehensive view on connecting WSN to the IP world and provides a lot of insight on mobility management for devices. The authors point out that key improvements to develop new efficient IoT systems are energy efficiency, security and operational autonomy of mobile nodes.

Roaming improvement might also come from enabling dual connectivity for a device. Thus [214] provides insight on various projects worldwide improve LPWA communications by plugging it into a satellite network as a backup link. [215] provides a complete comparative study of various LPWAN technologies, providing numbered information on mobility support, deployment cost comparison, packet loss, energy management, mobility latency. The study also put the solutions studied into perspective with other operated mobile networks (GSM, LTE) and provides insight on various possible use cases from asset tracking to healthcare. Vangelista et al. in [220] sums up recent LoRaWAN Standards evolution regarding roaming and other improvements. LoRaWAN is presented as the most promising candidate for worldwide interoperability and connectivity.

In this context, researchers have proposed various solution to mobility issues allowing for a user to access its data wherever his device is located. Most of them focus on gathering the user data using various solution, from data aggregation [224] [225] to publish/subscribe mechanisms [226] [227]. Researchers also looked for ways to trace devices in order to fit the network to their movements [228] [229] [230]. Finally improvements to the infrastructure were also studied from implementing a blockchain-based mechanism [231] [232], devices improvements were proposed [233] and interoperability solution were evoked [212] [4].

[224] uses aggregation and clustering to search services in a pool of IoT devices. It uses a semantic-based approach to explore heterogeneous networks using a decentralized approach to aggregate information fast and precisely. Their method theoretically ensures an accurate search system and shows good experimental results regarding time efficiency. In [225], Almajali et al. study mobile edge computing under the scope of mobility. They focus on highly mobile devices and solutions to improve device connectivity in unusual mobility cases.

[227] proposes a solution that consists in improving the MQTT to fit better to mobile scenarios. By managing data using an MQTT internal buffer, devices can re-deliver all their messages in the correct order even when some packets are lost or the connection is interrupted.

[226] explores how Software-Defined Networking paradigms might work in accord with a Data Distribution Service to create an efficient publish/subscribe mechanism as network backend to IoT infrastructures while addressing some key challenges to IoT infrastructures such as interconnectivity with an existing and standard network, mobility, service discovery and scalability.

[228] also builds an infrastructure based on Software-Defined Networking, which is used to divide the territory spatially in order to follow the devices and adapt the backend infrastructure based on its movements. Their solution also presents a global view on the network, which is acceptable for home networks but might present issues in a fully interoperable infrastructure with multiple actors and solutions. [229] uses Software Defined Networking to manage flows in order to efficiently forward data to a given location thanks to an efficient path estimator and flow manager. The solution tries to predict the path taken by a flow based on prior observations, thus reducing message overhead and energy consumption for flow tables. [230], with AFIRM (Adaptive Forwarding based Link Recovery for Mobility support) tries to use Named Data networks as a middle layer to support IoT mobility. Through an extensive study on routing, Meddeb et al. compare their solution to other approaches by studying data availability in a sensor mobility context.

[231] explores a different solution. Exploiting the capabilities of blockchain, the authors propose a fully decentralized LPWAN backend. Using blockchain allows building trust between networks, roaming agreements as smart contracts and resolving identities using a blockchain application. The paper builds a decentralized LoRaWAN architecture that would support roaming without building complex roaming agreements between partners. [232] also presents a blockchain-based infrastructure, building a federation using blockchain to enable truthful exchanges at the cost of around 0.5 to 2 ms loss in latency and which makes it hard for users outside of the federation to take control of the exchanges.

[233] creates a protocol called CoMP (CoAP-based Mobility Management Protocol) which keeps track of devices address and enable reliable data delivery when communicating. This solution can mitigate packet loss while keeping a short retransmission delay and handover latency.

In their work ([4]), Ayoub et al. exploit the newly-developed SCHC framework as a workaround to support roaming between LoRaWAN operators. They introduce an Application Mobility Service (AMS) in the LoRaWAN architecture which is contacted by the device using IPv6 after SCHC compression and is used to prevent uplink message repetition and improve session continuity.

2.6 DNS

2.6.1 How it works

DNS is a distributed lookup service used to translate between domain names and IP addresses. It already exists and is global. It is the most efficient, open and scalable system for name resolution. There can be no massive communication on the Web like email or web page resolution without DNS. At the beginning of the Internet, a simple host.txt file located on a single computer was responsible for the translation between IP addresses (such as 192.134.5.37) and domain names (Such as "www.afnic.fr"). As Internet grew Paul Mockapetris designed a more suitable distributed database (RFC 1034 [9] and RFC 1035 [10]) which is the DNS.

A host who wishes to access the web page of "www.afnic.fr" uses a client application such as web browsers or mail clients on the host computer. Such applications send a request to the local DNS resolver in the local Operating System (OS). The DNS resolver will invariably have a cache containing recent DNS lookups. If the cache can answer the request, the resolver will return the value in the cache to the application that made the request. If the cache does not contain the answer, the resolver will send the request to one or more designated DNS servers.

Assuming that the answer is not found in the local cache of the hosts computer, the resolver (client) sends a UDP recursive query to one of its configured local nameservers. In the case of home users, it will mostly be their Internet Service Provider(ISP). The local nameserver, in turn, looks for the answer in its local cache and, if an appropriate record is found, returns the cached address to the client.

On the contrary, if the answer is not found in the cache of the local nameserver, then the burden of finding the response for the resolver's query becomes the responsibility of the local name servers. The local nameserver queries the root nameserver for the address of www.afnic.fr. There are 13 root servers (from a.root servers.net to m.root servers.net). The root server process the query, and even though it does not

know the address of www.afnic.fr, it knows that the information should be under the control of the "Top Level Domain (TLD)" fr servers. In this case, one of the root servers will refer the query to the .fr nameservers. The local nameserver asks the .fr nameserver the same question and is referred to the "afnic.fr" nameservers. Finally, the local nameserver asks "afnic.fr" nameserver for the queried domain name address and gets the answer.

DNS has been studied since its first RFCs were published. [234], the reference paper within the ecosystem, first documented DNS effectiveness compared to its concurrent solutions, as well as DNS latency and DNS cache hit rate within a university campus. [235] provides additional insight regarding caching efficiency within a network, [235] also documents the quantity of DNS queries and responses observed compared to the overall traffic within their campus. [236] is a greater study that evaluates the responses from the .net and .com TLDs. The article also drafts a definition of a "normal" DNS resolver, leading them to study DNS "top-talkers", the 40000 resolvers that account for 90% of the traffic DNS in 2012. [237] studies DNS wrong configuration and its consequences (packet loss, NX domain, increase in latency ...) by studying various configurations (random sampling, zone transferred, 500 most popular web servers...) based on a custom DNS resolver.

2.6.2 Standards

The IETF is responsible for DNS-related standardization in RFCs. Initial discussion about having a structure such as the DNS started with [238] based on the domain concept. [239] discusses how the domain concept could be used for mail routing. [9] defines the concepts and [10] describes how the concepts could be implemented. There have been several RFCs focus on the type of Resource Records (RRs) used in the DNS starting with [240] which specifies the five new DNS types for experimental purposes, [241] for specifying the location services, [242] to specify the type on how the URIs could use the DNS and so on.

Another category of DNS standardization is on clarification of existing standards, such as [243] clarifying the initial DNS specifications in [9] and [10]. There are number of standards on DNS Operational issues such as [244] on common DNS data file configuration errors, [245] on common DNS operational and configuration errors, [246] on operational criteria for Root Name servers and [247] on use of DNS aliases for Network Services.

As per Bert Hubert [248] there are about 297 RFCs related to DNS. Hence it is impossible to provide an exhaustive list of all DNS related standardization documents. There have been even efforts from the DNS community to limit the number of Standards in the DNS [249]. The reason being with the growth in the DNS standards over the past three decades, it has become nearly impossible for DNS developers and users to read thousands of pages of standards work before any DNS related implementation could be done.

2.6.3 DNS Evolutions

As mentioned in the previous section [248], DNS is a vast domain in constant evolution from the first DNS standard published [238]. It should be noted that this section will not mention every modification related to the evolution of DNS; instead, we will focus on four specific aspects of DNS evolutions:

- the evolution to the transport of data over the network: DNS over DTLS, DNS over TLS, DNS over HTTPS and DNS over QUIC; that secures the link between the server and the user.
- the signature of DNS authoritative zone, with DNSSEC, which authenticates the integrity of the data sent from the server.
- the integrity check enabled by the use of DANE to countersign data and certificates
- the new paradigms introduced by the development of DNS-SD to improve local network self-configuration

All these aspects will be studied in the following part of the chapter.

Transporting DNS information securely

From the beginning of its use, DNS was mostly based on UDP connections, which allowed quick exchange between client and server without maintaining sessions or keeping track of packet losses . But DNS supported different possibility to transport its content from the first RFCs ([9][10]), as "queries are carried in UDP datagrams or over TCP connections" [9].

Thus DNS evolved to support additional transport layers, including more secure transport to encrypt its payload and prevent malicious third parties to listen to, intercept or modify DNS payloads. Its first notable development is the support for TLS encrypted channel usually called DNS over Transport Layer Security (DNS over TLS or DoT, RFC 7858 [250]) and its equivalent over Datagram Transport Layer Security (DNS over DTLS, RFC 8094 [251]). The work from RFC 7858 [250] aims to prevent eavesdropping from attackers as described in [252] by transporting both the DNS query and response over a TLS channel. The TLS connection was kept standard to support interoperability and stick to well-known implementations of the protocol. RFC 8094 [251], written around the same time, proposes to use DTLS implementations instead of TLS to prevent head-of-line blocking as well as to reduce the necessary handshake from TCP.

The issues behind DNS queries over TLS, which lead to the development of RFC 8484 [253], were twofold. On the one hand, it was noticed while developing DNS over TLS that the channel might be censored for technical or political reasons; thus, finding another way to transmit the information was deemed necessary to prevent "on-path devices from interfering with DNS operations". On another, as the most common use for the general public of the Internet is through a web browser, exploiting web technologies to share its channel consistently with Cross-Origin Resource Sharing (CORS) best practice and exploiting the capabilities of recent web browsers. Thus came DNS Queries over HTTPS (DoH), labeled RFC 8484 [253] to support these evolutions.

Another evolution today up for discussions is embedding DNS queries over QUIC. QUIC is a protocol developed by Google's team to improve the performance of their browser Google Chrome. QUIC relies on recent development and best practices from transports protocols to provide secure, fast and reliable encrypted communications over UDP. QUIC combines the low connection set up time and head of queue blocking from UDP with the retransmission efficiency and support from long messages from TCP while supporting the encryption and authentication introduced by

TLS or DTLS. The current version of the DNS over QUIC draft is accessible through ([254]).

These standards were also backed by performance measurements, tests and considerations by the research community. [255] studies the tradeoff from the loss in response time and packets that comes from securing and improving reliability when navigating the Web by comparing the response times from classic DNS (DNS over UDP or Do53), DoT and DoH. [256] provides additional information regarding DoT resolution with RTTs around 15ms for traditional DNS resolution and RTTs over 100ms for DoT use. They observe that securing DNS resolution with DoT comes with a 100ms tradeoff and multiplies the DNS response time by a 7-factor.

DNSSEC, signing a DNS authoritative zone

An important set of standards concerning DNS involves extending the DNS infrastructure for security purposes. DNS security extensions provide origin authentication and integrity assurance services for DNS data, including mechanisms for authenticated denial of existence of DNS data. [257] provides an introduction to DNS security and requirements, [258] defines the RRs for the Domain Name System Security Extensions (DNSSEC), [259] on the modifications required in the initial DNS protocols following the introduction of DNSSEC. [260] on storing certificates in the DNS, [261] on DNSSEC hashed authenticated denial of existence and [262] algorithm identifier allocation of DNSSEC.

DNS as an external integrity checker with DANE

Another important security extension envisioned with the DNS infrastructure is to allow X.509 digital certificates, commonly used for Transport Layer Security (TLS), to be bound to domain names using DNSSEC [263]. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA [264] standardizes a way to authenticate TLS client and server entities with and without a certificate authority. [265] precises acronyms to replace numerical values and simplify conversations about DANE, [266] Updates the DANE protocol and also provides Operational Guidance. [267] adds SMTP Security via Opportunistic DANE TLS, [268] adds the usage of SRV RRs with DANE and RFC 7929 [269] adds DANE Bindings for OpenPGP.

Discovering services using DNS

Standardizing Services discovery using DNS is the work of a specific group within the IETF called Extensions for Scalable DNS Service Discovery (dnssd) [270]. Born from the work from IETF Zero Configuration Networking (zeroconf) [271] working group, from Stuart Cheshire and Marc Krochmal's independent RFC DNS-Based Service Discovery (RFC 6763) [5] and upon the work from Apple Computer Inc.'s trademark, Bonjour algorithm. DNS-SD and hostname resolution aim to provide a framework for users and devices to automatize configurations, advertize services and connect to them.

The working group proposed two standards: DNS Push Notifications [272], and Discovery Proxy for Multicast DNS-Based Service Discovery [273], and three informational RFC: Requirements for Scalable DNS-Based Service Discovery (DNS-SD)

/ Multicast DNS (mDNS) Extensions [274], Selecting Labels for Use with Conventional DNS and Other Resolution Systems in DNS-Based Service Discovery [275] and DNS-Based Service Discovery (DNS-SD) Privacy and Security Requirements [276] to pave the way for possible implementations of DNS-SD applications from anyone on the market.

zeroconf working group published a single RFC [277] that paved the way for devices' self-configuration on the network. Their Dynamic Configuration of IPv4 Link-Local Addresses provides a method for hosts to automatically configure their IP interface with an IPv4 address for Link-Local communications. This solution uses the 169.254/16 IPv4 prefix registered for this specific purpose and is not routed on the Internet.

Then RFC 6763 [5] provides the first insight on using DNS for services discovery; it provides insight on structuring DNS resource records to facilitate service discovery. The document describes the necessary tools to deploy a self-configuration mechanism and support it within a local network, then describes a standardized syntax for devices to provide their identity and describe their services within the network. Finally, the document describes service discovery and how devices may access the data advertised by their network peers.

Evolutions to RFC 6763 were necessary to improve the solution's scalability, solve security concerns and extend the work from the RFC's author. Thus the dnssd working group came to provide these extensions.

RFC7558 [274] paved the way by providing the requirement to improve the scalability of DNS-SD extensions, and RFC 8222 [275] provided standardized labels to use when designing, advertising and researching a service using DNS-SD. RFC 8765 [272] is a first step to improve scalability by standardizing updates to a DNS-SD database, by updating a local DNS zone of authority and by reversing the standard use of DNS from its polling principle where users request information to a notification, publish/subscribe approach. RFC8766 [273] specifies a proxy mechanism that uses Multicast DNS to discover records and push them into a DNS namespace, thus providing a localized and centralized point within the local network that contains advertised services from other devices instead of asking them to listen to every advertisement and save them in their memory. This proves especially interesting in the IoT scope, where devices are not always listening to all communications.

DNS-Based Service Discovery (DNS-SD) Privacy and Security Requirements[276] provides advice and insight on implementing secure DNS-SD services. The RFC analyzes different scenarios, the RFC proposes various requirements to implement secure and privacy-preserving mechanisms to support DNS-SD-enabled communication. The document does not provide many solutions to its scenarios but asks all the questions necessary in a DNS-SD context regarding client and server security requirements to support new contributions in the DNS-SD community.

DNS-SD was also studied by the scientific community ([278], [279], [280], [281], [282], [283], [284], [285])

[278] listed the first requirements for DNS-SD in IoT applications. Requirements that were expanded and studied in a thesis [285] that proposed to connect the Smart Objects to the IP world using a combination of XMPP for messages and mDNS/DNS-SD for service discovery. [285], in his PhD defense, proposed various adaptation

layers for IoT, including a lightweight Bonjour implementation called uBonjour and a lightweight XMPP Stack called uXMPP2.

[279] developed a discovery protocol based on DNS-SD in Contiki OS and identified issues to solve in order to propose an efficient and interoperable service discovery for IoT. [280] also proposed such a solution for constrained devices based on Apple's mDNS/DNS-SD implementation and extended their work in [281] to propose scalability improvements to their solution to support large scale IoT deployments.

In [282], an extension to DNS-SD paradigms for Industrial IoT is proposed, and its reliability, efficiency and latency is tested extensively and show promising results. [283] proposed enhancements to the DNS-SD standard by exploiting the capabilities of DNS caches using techniques from the stateless DNS (sDNS) community. [284] continues the work on Service Discovery for IoT by leveraging the capabilities from DNS Name Autoconfiguration (DNSNA) implemented with a CoAP stack to support user configuration, monitoring and remote control.

2.7 Machine Learning and IoT

Artificial Intelligent approaches are efficient solutions to many issues, [144] is such an example where neural networks are exploited to support IoT specific issues such as improving compression capabilities by compressing data. Autoencoders are another interesting neural network approach to compression in IoT applications ([157])

Our work regarding Artificial Intelligent approaches for IoT applications mainly relies on improvements to compression techniques. A state-of-the-art on compression techniques, backed by such approaches, is provided at the beginning of 5 in section 5.2. This section will mostly summarize other Artificial Intelligent approaches related to IoT solutions to provide insight on possible applications of the techniques developed by researchers from this field to IoT solutions.

Leveraging Artificial Intelligent approaches in IoT contexts usually consist of two possible solutions: leveraging AI to exploit IoT data or empowering IoT devices using AI technologies.

An excellent example of leveraging AI to exploit IoT data is encountered in Industrial IoT use cases, for which sensor deployment can be massive and controlled and in which data can be centralized and easily exploited. [286], for example, exploits a framework build on cloud technologies to propose edge AI in an IIoT context, in which IoT data is pushed to the edge where data is processed. [287] reduces processing time in edge computing required in IoT applications by exploiting the resources from cloud infrastructures. [288] evaluates deep learning approaches in a combined IoT/Edge environment. [289] also exploits the edge infrastructure to improve IoT applications using a complex storage/processing system between edge and cloud to optimize all the tenants in processing IoT devices efficiently and with low latency.

Two other scenarios are usually studied: healthcare and agriculture. In [290], IoT devices are used to monitor a patient's heartbeat rate, and ML detects stress based on heartbeat data and enhances patient care based on its physiological condition. [291] proposes a combination of sensors, actuators and edge computing for smart farming with improvements in terms of resources scheduling compared to classic approaches. [292] and [293] build systems for water grid management that combines

sensors and actuators to manage the water infrastructure according to the needs detected by the combination of real-time measurements and predictions. [293] also exploits edge computing infrastructures to process videos, manager fertilizers supply and treat plant diseases and pests.

[294] exploits data collected in a museum from IoT devices to classify users behaviour to provide insight regarding customers to the museum stakeholders. [295] summarizes a lot of these approaches and provides a framework to exploit federated learning between IoT applications and exploit sensor data and behavior from other applications to kickstart IoT applications based on ML data analysis.

Many approaches for empowering IoT devices exploiting the strengths of AI consist of security enhancements, in [296] proposes to embark a ML component within an IoT gateway to detect traffic anomalies. [297] investigates attack models in IoT systems and present various machine-learning (ML) techniques for authentication or access control to protect users and data privacy. [298] analyses network traffic from IoT devices in a smart city environment to propose an Anomaly Detection system for IoT devices to detect compromised devices and mitigate attacks. [299] exploits ML capabilities to improve LoRa communications security and avoid security threats at the cost of scalability.

The LIMITS (LIghtweight Machine Learning for IoT Systems) [300] open-source framework is an interesting tool that proposes to validate and assist in integrating ML models in IoT applications. Such a solution helps developers and researchers to design embedded trained ML models onto sensors by estimating the feasibility of their project. Unfortunately, it cannot take into account framework-specific dependencies and keeps a generic approach.

Our approach differs from the approaches from this section as our work consists in embedding ML algorithms directly onto devices and realize ML-based calculations despite their constrained storage and processing power.

2.8 Discussion

Our readings regarding IoT, naming and its related applications lead us to work on various subjects. The general state of the IoT and its expected growth within the next years justify our approach to focus on interoperability and other performances challenges within IoT ecosystems, such as scalability or mobility support. These challenges lead us to study the possibility offered by querying mechanisms for IoT solutions, and their associated systems such as ONS [13], ORS or the Handle system.

The issue of querying mechanism is vast; thus, we focus on data storage, data retrieval, caching and discovery. Query performances and query approaches are also studied. Data storage usually rely on databases, from SQL to NoSQL, but also linked to other technologies, blockchain being one of them, the DNS another. Spatio-temporal data retrieval raise concerns regarding the location of databases and the cache expiration. Caching is a concern when working with mobile devices or changing information, for which cache duration, renewal or expiration are an operational focus. Various architectures can be discussed for querying; some are built within cloud approaches, while others rely on edge storage for lower response time. The DNS provides an efficient middle ground between cloud-centric solutions and edge approaches. A key necessity with querying is also discovery, various approaches

were studied, and DNS provides its own solution to the problem. Finally, we provide insight on query performances and approaches; recent work focused on new, innovative technologies, but DNS, as old as it is, remains the most used on the Internet.

Improving the scalability of IoT solutions might come with different approaches, we chose to focus on compression techniques to reduce payload size or network congestion. However, other approaches were mentioned, such as improving transmission scheduling or fine-tuning transmission parameters.

One of the major concerns within IoT is security. IoT solutions notoriously offer poor security qualities and rely on outdated authentication paradigms and cryptographic algorithms. Our focus regarding security relied on working with authentication, trust and privacy. We studied various authentication approaches for IoT, their strengths and weaknesses, from the research community and the SDOs such as IETF's work on authentication as part of the pana working group. Key concerns with trust are also addressed on building secure and efficient handshakes to bootstrap trust in IoT scenarios, which is a key concern for our work on IoTRoam. Finally, privacy preservation techniques are studied. In an all-connected world, privacy threats are a key concern to the security research community and evaluating the invasion of privacy and surfaces of attacks are essential to users' acceptance of IoT solutions.

Mobility management is one of the most important points when designing an IoT system. The devices studied within this thesis, such as sensors, watches, phones, luggage or cars, are highly mobile and supporting mobility is a key factor to improve IoT solutions. Mobility is, for example, a major concern in smart city environments. We studied mobility support through two aspects, coverage and roaming. Coverage is well documented; building solutions to improve coverage include developing new modulation techniques, developing new hardware or improving its knowledge of existing techniques to fine-tune the transmissions parameters. Roaming is a more organizational approach that rely on globally accessible RGs; within IoT ecosystems and its various network operators, roaming needs to rely on easy-to-deploy solutions, with collaboration inside an open ecosystem and communication in-between parties. Our focus on the LoRaWAN technology, and its open backend solution and specifications, is directly linked to this observation.

DNS, and its 297 RFCs, is a reference solution for information querying and discovery within the Internet. We detailed its functioning as well various DNS-linked standards that participate in the improvement of the global DNS ecosystem, including building trust through signed resources, securing transport channels and discovering resources.

Finally, we provided relevant information with regards to ML implementations for IoT ecosystems. ML is a popular subject that encompasses various approach section 5.2 will detail more specific information regarding compression techniques for IoT traffic. Thus, we provided insight into the way IoT solutions usually interact with the ML world. We observed that ML is often used to improve users' knowledge of their studied datasets or to support and complement existing solutions such as placement solutions or tracking solutions.

Chapter 3

IoTRoam, an IoT roaming federation

As presented in the previous chapter, mobility management is one of the most important points when designing an IoT system. We presented two possibility to support mobility of IoT devices:

- improving antennas coverage, by studying interferences, developing directional antennas or developing new modulations to support more device
- developing roaming between infrastructure, a more organizational approach that rely on collaborative work between operators that mutualize their infrastructure to reduce their costs.

Within IoT ecosystems and its various network operators, roaming needs to rely on easy-to-deploy solutions, with collaboration inside an open ecosystem and communication in-between parties. This chapter presents our work on improving Roaming capabilities in various IoT scenarios, with a focus on the LoRaWAN technology, and its open backend solution and specifications, using the DNS infrastructure as a facilitator. We describe our experimental platform and tests. Then present the consequences of introducing a distributed structure that satisfies the requirements of constrained IoT environments.

3.1 Introduction

Roaming is an ED's capability to transmit and receive data on a VN. A classic example is cellular roaming, wherein a subscriber can use the VN infrastructure (such as the radio spectrum, base station) - when the subscriber's HN does not provide coverage. Roaming between the HN and the VN needs to consider three broad criteria: technical, economic and regulatory. **Technically**, roaming requires an interconnection between the HN and the VN directly or via a third party. **Economically**, the interconnections between different network operators are governed by agreements, which define the terms of interconnection. There should be an external body (such as governments) which **regulates** these agreements so that the terms and conditions are beneficial both to subscribers and network operators. Our work focuses only on interconnection from the technical perspective.

Interconnection in IoT becomes possible either by establishing a direct **'One-to-One'** interconnection or using a **'Hub'** model. The 'One-to-One' interconnection is similar to the Internet peering model wherein two IoT networks interconnect. 'Hub' is similar to an Internet transit model, wherein by establishing an interconnection with a

single hub, it is possible to exchange traffic with the networks connected to that hub as well as with the networks connected to its peers. Both the 'Hub' and the 'One-to-One' interconnection models evolve as independent Silos wherein the ED in the coverage area of a VN can connect to its service only if there is a prior interconnection agreement between its HN and the VN or between the HN and the 'Hub'. The 'One-to-One' or 'Hub' interconnection deployments have been done following out-of-band mechanisms, and to our knowledge, there are no standardized interconnection procedures for interconnecting different IoT networks for roaming. In the independent silo scenario, when an IoT ED onboards to a VN, bootstrapping trust [301] is a crucial security concern. The ED needs to **be cryptographically authenticated** by the VN based on credentials such as its identifier and a PSK. Cryptography-based authentication usually relies on one or more trust anchors [2]. In the proprietary silo scenarios, the trust anchor information may be preset with the ED or established out of band.

We started with the idea of setting up an **open roaming federated platform** integrating all IoT connectivity technologies. We debated this idea by discussing it with the IETF IoT onboarding mailing list [302]. This discussion made us realize that we should focus on a specific IoT connectivity technology and, if possible, extend the concept to other IoT technologies. IoT connectivity technologies could be classified broadly into three categories [15]: short-range (Bluetooth, Zigbee, Zwave), medium-range (Wi-Fi) and long-range (LoRa, NB-IoT, Wi-Sun, Sigfox). We eliminated from our focus technologies that cannot support roaming, such as Short-range technologies and closed networks such as Sigfox, which do not require the roaming feature, as it is a single network. Narrowing our focus based on requirements, we short-listed LoRaWAN [303], which falls under the LPWAN Technologies [304] category. Compared to other IoT connectivity technologies, the LoRaWAN ecosystem provides freedom to its stakeholders to choose the ED manufacturers, network service providers and application service providers. Since the radio connectivity uses a license-free spectrum, the freedom of choice in LoRaWAN also extends to deployment options. There are **public** LoRaWAN having nationwide coverage; **private** LoRaWAN focusing on specific use-cases and **community** networks that can be used for free by end-users.

We intend to test the federated platform with academic institutions as an open and accessible interconnected IoT network without considering economic factors. We started roaming testings using this platform with research and academic institutions. This initiative was presented to the LoRa Alliance (the SDO responsible for LoRaWAN specifications) and will be included as part of their academic outreach program.

Architectures proposing solutions to the technology barriers mentioned earlier should consider the constrained characteristics of IoT environments. If the proposed architecture is validated with LoRaWAN having constraints such as the maximum frame size of 51 bytes (or 222 bytes for lower SFs) and latency requirements of two seconds for default uplink/downlink exchange, we hypothesize that architecture is extendable to other IoT networks. Mobility between the three different types of LoRaWAN networks (public, private, community) is a significant issue. A company may use LoRaWAN to monitor the battery level of vehicles in its fleet, an agricultural cooperative may use LoRaWAN to monitor the stock flows of its associates, or an emergency service may use LoRaWAN to coordinate its teams

in the field. Most existing studies on LoRaWAN consider scenarios where the EDs are mobile, but remain under the umbrella of the same NS [305].

We labeled this platform **IoTRoam** and its objective is to achieve, with IoT connectivity technologies the same interconnection functionalities that eduroam [306] proposes with Wi-Fi connection. In eduroam, an end-user who has credentials to connect to a particular eduroam Wi-Fi network for Internet access can access the Internet from any other eduroam network seamlessly. The **first requirement** is that an ED having credentials to connect to a particular IoTRoam network should be able to access its service seamlessly (with minimum prior configuration requirements) in the event of finding itself in the coverage area of the VN. The **second requirement** is that the proposed federated model should be operationally feasible. The vision is to start with LoRaWAN and extend the design to apply them to other IoT networks. The IoTRoam architecture aims to enable interoperability between the silos in the IoT domain by leveraging the DNS protocol, its security extensions (DNSSEC [307]) and a PKI using self-signed X.509 digital certificates, thus bringing in the following **contributions**:

- The proposed architecture enables **roaming between different LoRaWAN networks** without the need of having any prior interconnection agreement
- The architecture includes an **AA framework based on PKI** enabling **secure onboarding** of the IoT ED;
- The architecture satisfies basic IoT operational requirements such as **scaling**, viability by **not incurring additional costs**, **ease of deployment**, **interoperability** between different IoT networks involving multiple stakeholders;
- Experiences from the implementation as a Proof of Concept (PoC) has enabled us to propose **three** accepted change requests (Change request is the procedure to provide modifications to the LoRaWAN specifications);
- With this PoC, we tested different LoRaWAN roaming scenarios with two Institutions in France - IMT Atlantique and Telecom SudParis (TSP). We ran measurements to assess whether the additional overhead introduced by the proposed architecture meets the constrained requirements of LoRaWAN;

Accounting was kept out of the scope of this project but devising device classes and monitoring roaming user's band use could be interesting to study afterwards. IoTRoam's added value is the possibility of using core Internet infrastructures such as DNS and PKI to enable interconnection and security of IoT ED onboarding. The objective is to extend Internet resolution and security infrastructure services to be adapted to IoT, thus enabling seamless interoperability.

The remaining parts of this chapter are structured as follows: Based on the literature, Section 3.2 identifies the requirements for a secure and seamless interconnection architecture. Section 3.3 summarize LoRaWAN key aspects regarding authorization and authentication, section 3.4 presents our design choices and section 3.5 describes our integration of PKI paradygmes in the network. Then section 3.6 describes our experimental setup to validate our proposed architecture for LoRaWAN passive roaming. In Section 3.7, we evaluate whether the proposed mechanisms satisfy LoRaWAN constraints. We propose to extend this solution by provisionning the DNS queried information beforehand using a combination of prefetching techniques and mobility prediction algorithms (3.8). Finally, we sum up our contributions in 3.9 and conclusion in 3.10.

3.2 Motivation for a federated interconnection model

Since the focus of the chapter is on interconnecting IoT networks, our initial approach was to reuse existing standardized Interconnection models for roaming. The hurdle is that there is no single SDO that has the sole responsibility of making IoT Standards. As to our knowledge, there is no standardization work on IoT interconnection for roaming, satisfying the following basic roaming requirements:

- The VN should be able to **provide the roaming service** even if it does not have a prior interconnection agreement with the ED's HN;
- The VN should be able to **control the terms** under which a roaming ED is allowed to use its resources securely;
- The infrastructure interconnecting the home and the VN should be **able to scale**;
- The interconnecting infrastructure should be **open**, **global** and **viable operationally**;

A recent research study [308] proposed a mechanism to enable roaming between LoRaWAN and 5G network. This proposal includes a handover roaming mechanism for LoRaWAN that relies on 5G to authenticate the ED to the network. To our knowledge, as part of the LoRaWAN community, tests have been done for passive roaming, but handover roaming scenarios are still in the pipeline. Also, in the article, the ED is intended to be equipped with both 5G and LoRa interfaces. Adding a 5G interface to the ED will considerably reduce the battery life, which is a disadvantage when considering operational feasibility. Keeping our focus on building an operationally feasible interconnected IoT platform, we turned to the WBA Open Roaming [309] initiative for guidelines. Concerning the basic requirements for designing an architecture for an Open, seamless IoT interconnection, a WBA study [310] outlined a set of requirements to consider.

When an IoT ED is roaming, the VN should retrieve its identifier from the incoming Join Request (JR) packet to identify the ED's HN. Therefore, identifiers play a vital role in IoT interconnection [311] [312]. IoT identifiers are structured into two different categories: **Hierarchical** and **Flat**. An example of a hierarchical identifier is EPC [313]. The barcodes attached to consumer products are based on EPC identification. An example of a flat identifier is UDID, a unique serial number assigned to track and record each Apple manufactured device. Both Apple and the EPC identity management infrastructures use closed databases to provision the identifiers. Mapping the ED's identifier to its appropriate network or service is only possible for entities who are provided access to these databases. From a global (not just limiting to LoRaWAN) IoT perspective, the **first issue** to resolve is to let different IoT sectors use their existing identifiers but to use a global database for IoT allocation and resolution.

The **second issue** is to use a global AA model, which controls the terms under which a roaming ED is allowed to securely use the resources in the environments operated by the VN. AA functionalities are usually consolidated in a single centralized database [184]. The centralized AA framework has its advantages and significant disadvantages, such as creating a single point of failure. Blockchain using distributed ledger has been experimented with and deployed [185][186] to accomplish

a scalable decentralized AA framework. Nevertheless, the blockchain model has several drawbacks as a feasible operational model [187] in an open/global scenario.

A **third technology barrier** that we consider is that any proposed architecture should satisfy IoT environment constraints requirements detailed further. Narrowing our focus on requirements, we short-listed LoRaWAN due to its open standard characteristics and its ability to set up a private roaming set up.

The DNS infrastructure serves, among many other uses, to link domain names and IP addresses and is **scalable** and **operationally viable** on the Internet. Standards such as ONS [13] for the consumer industry, Object Resolution System (ORS) standardized jointly by the ITU-T and ISO/IEC, and the Handle system standardized by the ISO uses the DNS infrastructure to resolve the IoT identifiers to its related service on the Internet. DNS has been used by Mobile Network Operators (MNOs) on the inter-operator IP backbone network to enable data roaming [314].

eduroam [306], a Wi-Fi-based roaming platform widely adopted in the academic community, uses a distributed PKI based on X.509 digital certificates for AA. The trust fabric in eduroam is a PSK between the RADIUS servers (organizational, national, global) based on the DNS hierarchy. Such a trust fabric, wherein a PSK is shared hierarchically, hinders the design that we envision for IoTRoam. Different IoT networks use different mechanisms to share the PSK between the ED and the AA servers on the Internet to securely onboard the ED to its HN. Forcing them to transition to a newly proposed PSK mechanism is not operationally possible since multiple stakeholders are involved. We proposed to use the PKI based on X.509 self-signed digital certificates and DNSSEC trust anchor fabric that allows the IoT stakeholders to use their existing PSK mechanism.

IoT Roaming is different from cellular roaming and thus posing new **challenges**. In cellular roaming, interconnection is usually geographically defined and shared between different public MNOs (usually three to four MNOs in a country). In the IoT scenario, there are public, private, community-based network operators, and there could be thousands of private network operators within a Country. Adding to this complexity, IoT roaming mostly needs to have multi-layer interconnection agreements. For example, the authentication and authorization of the ED to the roaming network may be governed by a security solution provider or by the ED manufacturer rather than the network operator.

The IoTRoam experience brings the following **contributions**:

- A model that seamlessly **interconnects** the multi-stakeholders, IoT connectivity technologies using standards and infrastructure currently employed on the Internet
- A model that is **operationally feasible** and can be deployed with **minimum prior configuration requirements**
- A **PoC**[315] in place, which is **open**, can be **accessed freely** and used by the **community**. In this process, we have also contributed to software developments [316]

3.3 LoRaWAN Interconnection with regards to Authentication and Authorization

LoRaWAN is an asymmetric protocol, with a star topology as shown in (Figure 3.1). Data transmitted by the ED is received by a RG, which relays it to a NS. The NS decides on further processing the incoming data based on the ED's **unique** identifier (DevEUI). The NS has multiple responsibilities like forwarding the uplink from the ED to the AS, queuing the downlink from the AS to the ED, forwarding the ED onboarding request to the appropriate AA servers, named as Join Server (JS) in LoRaWAN terminology. The AS handles all the application-layer payloads of the associated EDs and provides application-level service to the end-user. While the ED is connected to the RG via **LoRa modulated messages**, the connection between the RG, the NS and the AS is done through **IP packets** and can be backhauled via Wi-Fi, hardwired Ethernet or Cellular connection.

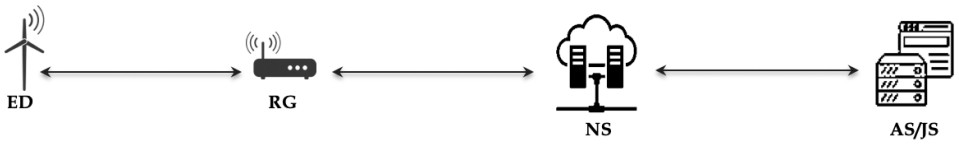


FIGURE 3.1: Basic LoRaWAN set up

The JS acting as the AA server control the terms on how the ED gets activated (i.e. onboarded) to a selected LoRaWAN. There are two types of ED activation: Over the Air Activation (OTAA) and Activation by Personalization (ABP). With ABP, the ED is directly connected to a LoRaWAN by hardcoding the cryptographic keys and other parameters required for secured communication, Roaming would not be possible in this first case. With OTAA, the parameters necessary to create a secured session between the ED and the servers on the Internet are dynamically created for a session. This is similar to TLS handshake. OTAA is preferred over ABP since it is dynamic, decouples the ED and the backend infrastructure and does not need configuration parameter hardcoding. This chapter will focus only on the OTAA process. In the **HN scenario**, the ED performs a Join procedure with the JS during OTAA by sending the JR. The JR payload contains the ED's unique identifier (**DevEUI**), the associated application identifier (**App EUI**) and **JoinEUI** (unique identifier pointing to the JS).

The JS associated with the ED also has prior information such as the ED's DevEUI, the cryptographic keys: NwkKey and AppKey required for generating **session** keys to secure the communication between the ED and the NS and AS. These are the pre-shared information between the ED and JS (the AA server) on the Internet, which we proposed not to modify. Once the JS authenticates the ED, it responds with a JoinAns message to the NS. The JoinAns message contains different session keys derived from the root keys: one set of cryptographic keys for securing the *ED* <--> *NS* interface and another for securing the communication between the *ED* <--> *AS* interface, for a particular session.

In the **VN scenario**, a **non-activated** ED should first activate itself and then transmit/receive the payload. Roaming scenarios in LoRaWAN are classified into **passive**

and **handover** roaming. We limit ourselves to passive roaming since handover roaming is still in the testing stage and an open LoRaWAN software stack for handover roaming is not yet available.

In **passive roaming**, the MAC layer control of the ED is maintained by the home NS, which becomes the serving NS (**sNS**), as shown in Figure 3.2. The roaming ED uses the NS of the VN named forwarding NS (**fNS**) to send messages to its sNS. The fNS forwards messages between the sNS and the ED. If the ED is not yet activated, then it has to get activated using **passive roaming activation** process as shown in Figure 3.2. When the fNS does not have prior information about the sNS, the fNS SHALL use the DNS to find the roaming ED's JS IP address.

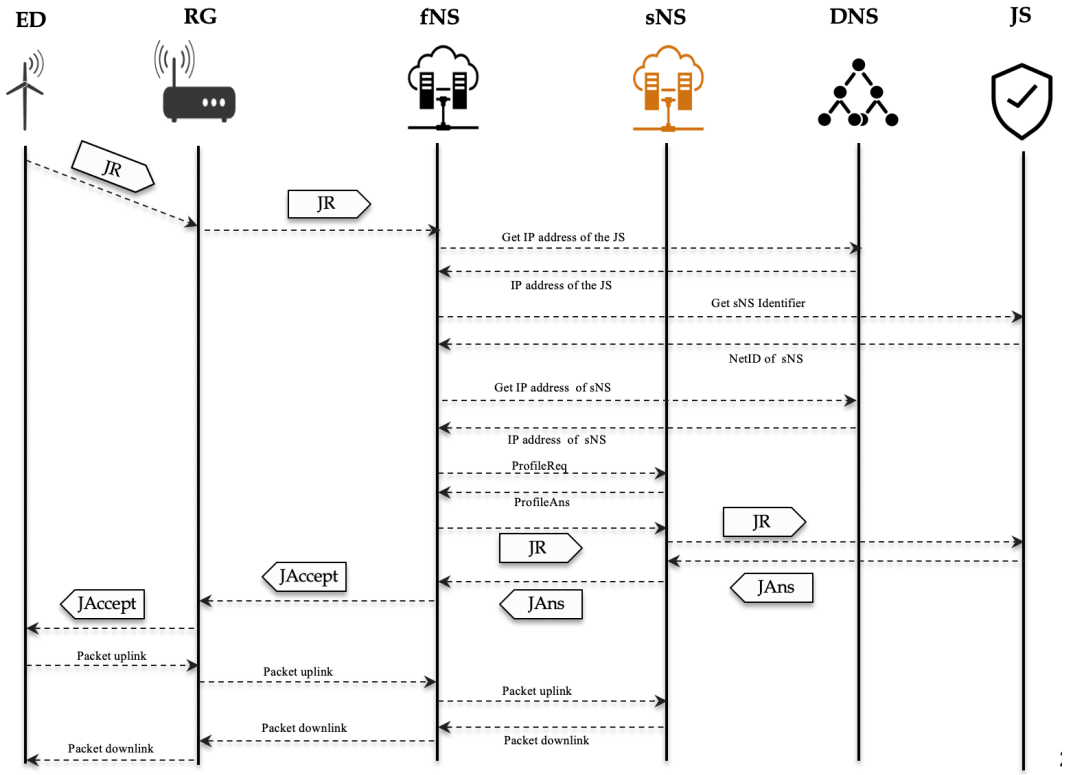


FIGURE 3.2: Passive Roaming Activation message flow

As per the LoRaWAN backend specifications, the LoRa Alliance has allocated a DNS Zone file (**joineuis.lorawan.net**) for provisioning the information mapping the JoinEUI to its corresponding JS operator. Each nibble of the JoinEUI represented in the hexadecimal format "**0x00005E100000002F**" is first reversed. Then periods are inserted between each nibble and the domain name **joineuis.lorawan.net** is concatenated as the suffix (JoinEUIs are theoretically hierarchical values based on IEEE OUIs tables). The final result is a domain name that can be provisioned in the DNS zone file **joineuis.lorawan.net** pointing to their respective JS as follows:

f.2.0.0.0.0.0.0.1.e.5.0.0.0.0.joineuis.lorawan.net. IN A 192.168.1.1

Thus the fNS, by running a standard DNS resolution, can retrieve the IP address of the JS corresponding to the JoinEUI and kickstart the **Join procedure**. The fNS then queries and obtains the **NetID** (i.e. the 24 bit Unique Network Identifier of the sNS represented in the hexadecimal format: "0x60050A") from the JS. Similar to

JoinEUI, the LoRaWAN backend specifications has allocated a specific DNS Zone file (**netids.lorawan.net**) for mapping the NetID's to their corresponding NS. Any LoRaWAN operator(either private, public or community) needs to obtain from the LoRa Alliance a unique NetID (which is a flat identifier) and provision it in the DNS zone file "**netids.lorawan.net**", a standardized DNS resource record pointing the allocated NetID to its sNS as follows:

60050a.netids.lorawan.net. IN A 192.168.1.2

Thus for a fNS, it is possible to resolve the sNS of an ED by querying the NetIDs DNS zone file, even if there is no prior roaming agreement. The sNS and the fNS exchange data, and finally, the ED is activated once the ED receives the JoinAccept (JAccept) with the session keys for transmitting uplink or downlink messages. The DNS provisioning mechanism has ensured that both JoinEUI and NetID could be provisioned or updated by different entities in their respective DNS Zones (Servers); they are unique in the global scope and cannot be duplicated. The DNS resolution mechanism ensures that both the JS and the NS can be accessed from anywhere on the Internet with a simple DNS resolution. Figure 3.2 demonstrates how multi-stakeholders interconnection complexities are solved due to DNS provisioning and resolution since for a single ED onboarding, the JS could be operated by a different entity than the NS operator. Thus, the LoRaWAN architecture design by itself provides a **partial** solution to the WBA requirements described in section 3.2.

3.4 Design choices regarding IoT identifiers provisionning

Both previously mentioned IoT identifier types, hierarchical (EPC) and flat (UDID), could be accessed from the global Internet if they are provisioned in the global DNS database (Figure 3.3). Then it is up to the client libraries to make the conversion and add the specific sub-domain suffix (udid.apple for UDID and gs1.fr for EPC) to the identifiers. Once the identifier is converted to a domain name as follows:

2b6f0cc904d137be2e1730235f5664094b831186.udid.apple.
3.1.3.1.6.2.3.3.9.3.4.0.3.gs1.fr.

It will follow the normal DNS resolution process to resolve the identifier's associated resource/service/metadata globally.

Some parameters such as ED's HN identity, the AA server identity, the authentication credentials and the port numbers must be configured in proprietary roaming models such as a hub before an ED can roam outside its HN. Except for the authentication credentials, all other information could be retrieved from the DNS database. Thus, by provisioning their IoT identifiers and related information in the DNS database under their own domain namespace, different IoT sectors could interoperate by using their existing identifiers, thus satisfying operational viability. The ED is configured with a PSK that is only shared with an AA server, creating the session keys for encrypted communication between the ED and the different associated servers on the Internet. When the ED is onboarding in a VN, the VN should establish mutual authentication with the ED's AA servers and the HN. To establish mutual authentication dynamically between different servers on the Internet managed by multiple stakeholders, our hypothesis is to use the DNSSEC infrastructure as trust anchors and a PKI based on self-signed X.509 digital certificates. The DNSSEC extensions use asymmetric cryptographic signature mechanisms to authenticate the

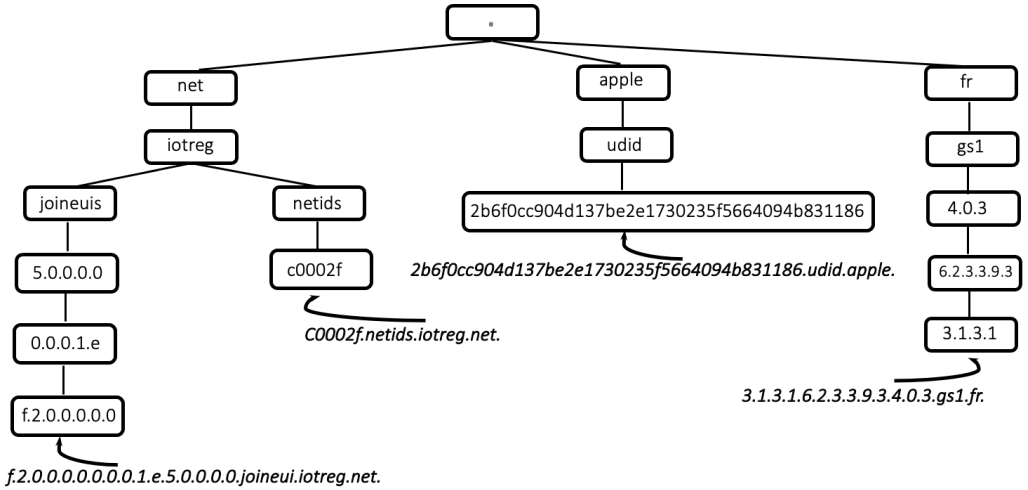


FIGURE 3.3: Provisioning IoT identifiers on the Internet domain namespace

data provisioned in the DNS database. The Signatures and public keys come in the form of new DNS records that provide authentication. With DNSSEC, the origin and integrity of received data can be verified using one or more key pairs associated with the DNS zone.

DNS is a time-tested infrastructure and had scaled from hundreds of domains from the Internet's beginning to billions currently [14]. These factors influenced our choice to use the DNS infrastructure, its security extensions and a PKI in the LoRaWAN roaming architecture. A DNS infrastructure, similar to the LoRa Alliance's **lorawan.net**, was set up under the domain **iotreg.net** for provisioning the JoinEUIs and NetIDs, as shown in Figure 3.3. Each nibble of the JoinEUI represented in the hexadecimal format **0x00005E100000002F** is first reversed. Then, periods are inserted between each nibble and the domain name joineuis.iotreg.net is concatenated as the suffix. The final result is a domain name provisioned in the DNS database pointing to their respective JS as follows:

f.2.0.0.0.0.0.0.0.1.e.5.0.0.0.0.joineuis.iotreg.net. IN A 192.168.1.1

Similar to the JoinEUI, the NetID represented in the hexadecimal format are provisioned into the DNS without reversing and adding periods between each digit, pointing the allocated NetID to its NS is as follows:

c0002f.netids.iotreg.net. IN A 192.168.1.2

The JoinEUI is reversed, and periods are added since it benefits from a hierarchical model and the NetID is based on the flat model. The DNS provisioning mechanism has ensured that both JoinEUI and NetID could be provisioned or updated by different entities in their respective DNS Zones; they are **unique in the global scope** and **cannot be duplicated**. Both JS and the NS can be accessed from anywhere on the Internet, and with a simple DNS resolution, the JoinEUI can be resolved to its JS and NetID to its NS dynamically without any prior interconnection agreements shared in advance. The JS and the NS DNS resolution information are secured from being spoofed on the wire and modified at the DNS database since the DNS infrastructure is signed by DNSSEC. We developed and provided a secure, automatized

DNS provisioning platform that could be used by the community. With a secured API key, any authorized user can access the User Interface (UI) (via web or API). The UI enables authorized users to do multiple operations (creation, modification, deletion) of only their data in the DNS database. To make it easy for the community to understand and use the interface, a video tutorial [317] is provided. While testing the UI with some LoRa Alliance community members, we encountered operational issues such as validating that the data provisioned in the DNS is done by the rightful owner. The need to validate the JoinEUI (an IEEE EUI-64 identifier provisioned by the IEEE and has Organizational Unique Identifier (OUI) in the IEEE EUI-64) with the IEEE OUI database, were identified and implemented, thanks to the PoC. The implemented solution has been provided as feedback to the LoRa Alliance, which could be integrated when the DNS service operated by the LoRa Alliance is deployed.

There was no existing off-shelf or open-source LoRaWAN network stack software that uses DNS for ED onboarding or roaming. We collaborated with the open-source Chirpstack network stack [318] author to update the software to integrate both functionalities. The NS, JS and the AS in our PoC are installed with appropriate software from Chirpstack, thus enabling DNS resolution.

3.5 Security integration to the experimental set up and validation

For secure ED onboarding, the interface between the servers (NS, JS and the AS, which could be grouped as **backend elements**) in the **IP space** (Figure 3.5) should be mutually authenticated (i.e., both the client and the server authenticate each other), as per the LoRaWAN Backend Interface Specification [303]. However, the mechanisms for mutual authentication is left to the implementer's choice and is not normative.

The PKI using the X.509 digital certificates signed by a **trusted Certificate Authority** (CA) is widely used to secure web traffic. However, the CA trust model for issuing the X.509 digital certificates is not operationally feasible for IoTRoam. On the web, the browser client (such as Chrome, Firefox) has a certificate store containing thousands of Root CA certificates. The browser authenticates any server that delivers a X.509 certificate digitally signed by anyone of the Root CA in its certificate store. Such certificate store infrastructure is not available in the LoRaWAN backend network elements or any IoT backend infrastructures. Even if we assume the infrastructure exists, the digital certificates come at a cost, which is not viable for most IoT services. We tried with Let's Encrypt, which provides X.509 digital certificates for free. However, it was not possible to benefit since they do not provide certificates for domain names with more than ten labels (JoinEUI has more than 16 labels). A viable solution to resolve the operational and cost issue is to generate our Self-Signed certificates.

Our certificate provisioning model is that any institution willing to test roaming based on the IoTRoam architecture can request intermediate certificates from a trusted **Root CA**. Figure 3.4 shows a scenario wherein Afnic plays the role of Root CA and generates intermediate certificates for two independent LoRaWAN networks - TSP & Afnic Labs. The intermediate CAs will, in turn, generate the leaf certificates for backend elements. Details on obtaining an **intermediate certificate**

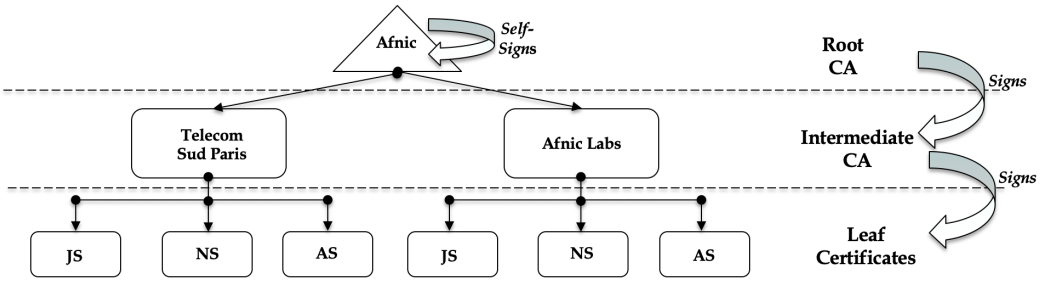


FIGURE 3.4: IoTRoam Certificate provisioning infrastructure

and generating the leaf certificates are documented [315]. We further simplified the process, wherein the institutions can generate the leaf certificates by just running a makefile after customizing their JSON configuration files and adding the provided leaf certificates information into each of the backend elements configuration files.

3.6 Experimental Setup

To validate the architecture, two independent LoRaWAN networks were set up separated by a distance of 34 kilometres. The two locations are Afnic (in the Yvelines department in France) and Télécom SudParis (in the Essonne department in France). The backend elements are installed with the open-source Chirpstack network stack and are configured with their respective intermediate and leaf certificates.

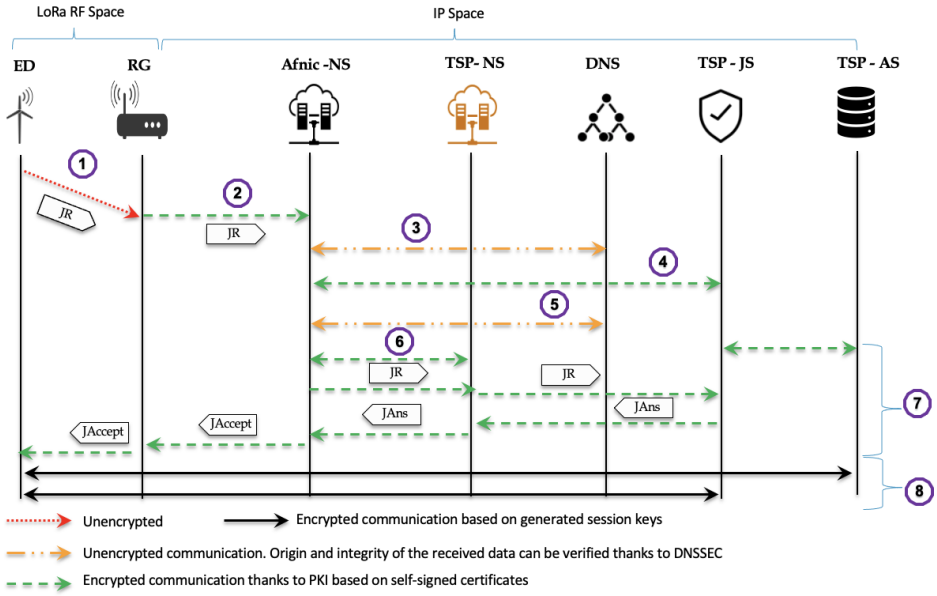


FIGURE 3.5: Testing Passive roaming ED onboarding using the proposed architecture

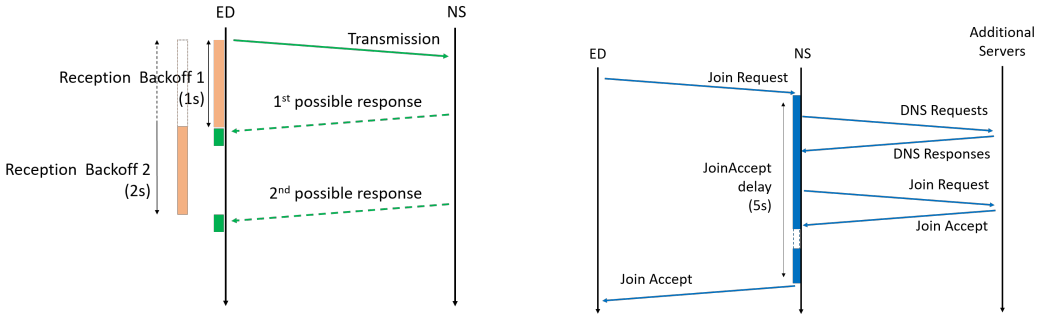
Figure 3.5 shows that the ED configured with TSP as HN uses the RG in Afnic's coverage area to onboard (Step 1). The RG forwards (Step 2) the incoming JR to the Afnic NS, which in turn uses the DNS infrastructure (Step 3) to retrieve the TSP-JS IP address (based on the JoinEUI in the JR) since the ED is unknown to it. Afnic-NS and the TSP- JS runs a TLS handshake for mutual authentication (Step 4). During mutual authentication testing, we identified that combining the intermediate and the server

leaf certificate (a combined trust chain) during a TLS handshake could bypass the need for having a certificate store with all intermediate certificates and store only the Root CA certificate. The certificate validation process is done by sending the combined trust chain to the server's IP address. On receiving the combined trust chain, the server first verifies the leaf certificate in the combined trust chain. When the leaf certificate is unknown, it checks the following certificate in the chain, the intermediate certificate. Since the Root CA signs the intermediate certificate, the combined certificate chain becomes trusted. Thus, the backend network elements (NS, AS and the JS) could be mutually authenticated even if they are in different networks since they have a common Root CA at the top of the chain of trust.

On a successful mutual authentication between the Afnic-NS and the TSP-JS, Afnic-NS retrieves the NetID of the ED from the TSP-JS (Step 4). Using the retrieved NetID, the IP address of the ED's NS (i.e., TSP-NS) is obtained (Step 5) via DNS resolution, and mutual authentication is established between the Afnic-NS and TSP-NS (Step 6). Once the mutual authentication is established between the different servers in the IP interface, the JR is sent to the TSP-JS to create the cryptographic session keys. The cryptographic session keys are sent back to the ED via the PKI secured mutual authentication channel as Join Answer (JAns) and Join Accept (JAccept) (Step 7). Finally, a secured session between the ED and the associated servers on the Internet using the generated session keys (Step 8).

3.7 Performance evaluation

The time taken for the ED to onboard (i.e. Steps 1-7 in Figure 3.5) is the metric that we want to measure to study the latency influenced by DNS and PKI. In the LoRAWAN terminology, the onboarding process is termed as OTAA.



(A) Uplink/Downlink reception window in LoRaWAN

(B) OTAA Reception window in LoRaWAN

FIGURE 3.6: Reception windows in LoRaWAN

Following an uplink, a Class 'A' ED in LoRaWAN opens a **receive window** for one second (default value), and if no downlink is received during the period, it opens a **second receive window** after another second (default value) as shown in the figure 3.6a. If no downlink communications are received from the server between the two-receiver window, it must wait until the ED triggers the next uplink and opens a receive window. For the ED onboarding process (i.e. OTAA), in the EU 868 Mhz channel, the default Join Delay window, as described in [304], and illustrated on Figure 3.6b, is **five seconds** meaning the RG will transmit the downlink JAccept exactly five seconds after the uplink.

The performance evaluation objective is to check whether the introduction of DNS and PKI influences the onboarding process time. We defined three scenarios for our measurements:

- Scenario 1: The ED is in the HN without the latency introduced by DNS or PKI
- Scenario 2: The ED is in the HN, but the NS and JS are resolved using DNS resolution
- Scenario 3: The ED is in the VN's coverage area with the latency introduced by DNS and PKI for mutual authentication

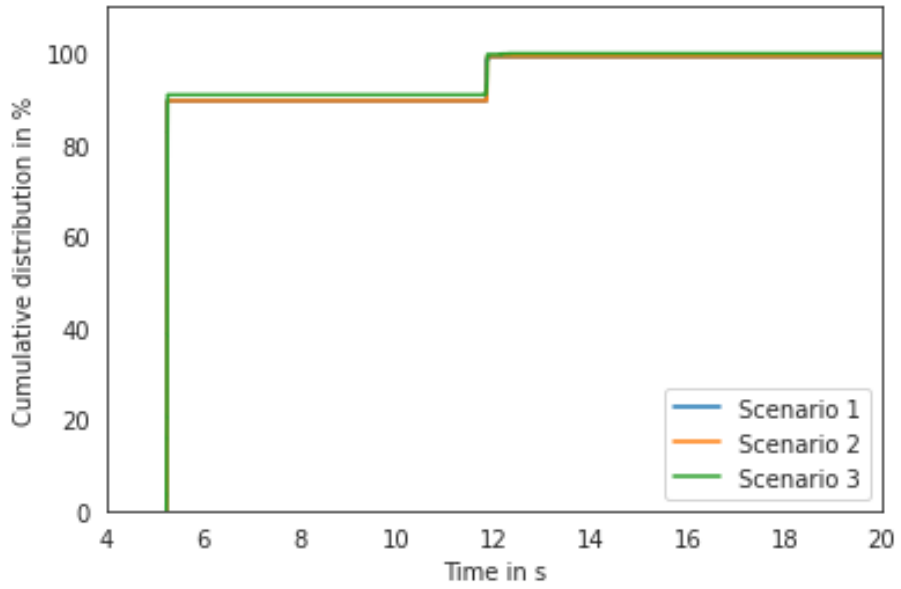


FIGURE 3.7: Cumulative distribution of the ED onboarding delay measured on the ED in ms

To ensure that the measurement is precise and get rid of any synchronization error between the ED and the backend network elements, the measurements were realized directly on the ED. We ran the measurements for around **30 hours of transmissions** and gathered more than 2000 measurements for each scenario.

Figure 3.7 shows the time-to-join for the three scenarios obtained by monitoring the delay between the JR and the "Join Success" message received at the ED. In the EU 868 Mhz channel plan for LoRaWAN, the Join Delay window is 5 seconds [304] meaning the RG will transmit the downlink JAccept exactly 5s after the uplink (Figure 3.6b). The RG may receive the downlink JAccept well in advance, but it will stay in the queue until the requested TX time. This means that the ED will receive the JAccept after 5s. Our measurements show that the device receives its JAccept around 90% of the time in all scenarios, **around 5.2s after sending its JR**. The ED can onboard as soon as possible independently of using DNS or experimenting with a roaming ED. Therefore, DNS seems to have no significant impact on the activation delay. A fact that can be explained considering that the ED's Join Delay is significantly lower than the standard times for DNS resolutions (usually around 300ms).

Figure 3.8 shows the first delay for end-to-end communication after activation. Once

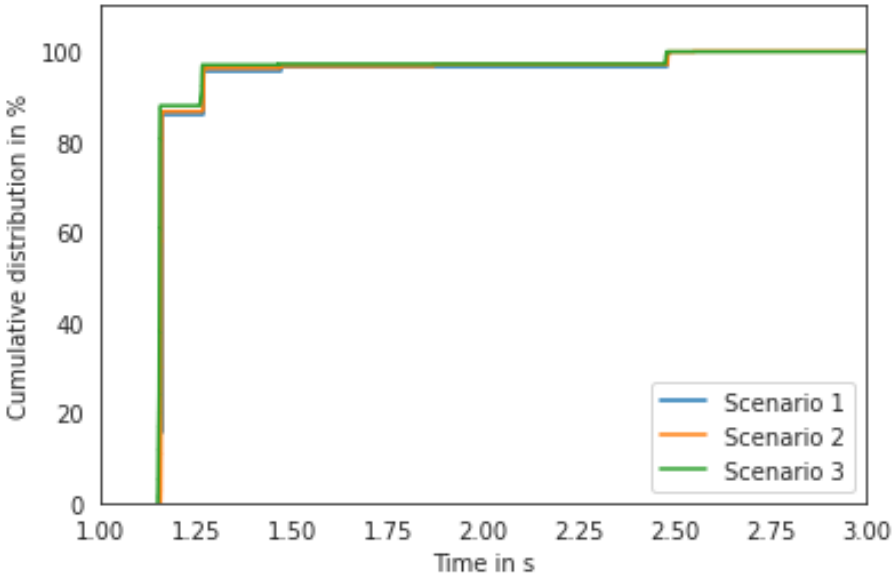


FIGURE 3.8: Cumulative distribution of the first uplink delay measured on the ED in ms

again, we see that our **data is gathered around two values**, 1.2s and 2.4s, which correspond to the **two receive windows** available when class A LoRaWAN ED communicate and illustrated on Figure 3.6a. The ED can receive the acknowledgement within the receive window's time limit regardless of the scenario studied.

These measurements lead us to believe that introducing DNS and PKI to the LoRaWAN system would not significantly add to the latency in LoRaWAN communications. It is of note that our measurements show no significant packets losses as our testbed is quite optimal. Additionnal measurements varying interferences should be of interest to enhance this work. Also, we configured our infrastructure to work without DNS caching. A regular infrastructure would **further benefit from DNS caching** as a way to reduce the impact of DNS[234]. However, we propose to expand our reflection on the impact of DNS by studing possible **prefetching and caching scenarios for DNS data** in a mobile environment.

3.8 Prefetching of mobile devices information to reduce DNS impact

The DNS is a key component in IoTRoam, as a focal point to support roaming, it might incur additional latency and hinder device connectivity. From a user's point of view, it is important that access is provided as smoothly as possible, **without additional cost**, to develop the technology's adoption. And in roaming scenarios, serving all users as soon as possible would decrease the impact from other networks on their own gateways. Thus, reducing the impact from DNS requests when a device is joining becomes a **key connectivity concern**. From an operator's point of view, increase in latency might incur **congestion** or **gateway overload** which would decrease the QoS for IoT solution.

Beyond our use case, the issue behind storing and sharing DNS data, or where to locate a DNS cache, how long to keep information cached and when to access it as

an operator, is crucial to improve network for mechanisms such as those presented in Chapter 4 and 5.

Prefetching information is a common strategy to reduce latency within networks. Web browsers make use of such techniques to obtain IP addresses for domains within a web page, predicting that the user may click on a link, thus sparing the DNS requests when a user clicks by performing the request beforehand.

DNS prefetching relies on a prediction mechanism; the user could click on the link, so its browser performs the query beforehand. This simple prediction mechanism can be applied to any circumstances. [319] analyzed DNS traffic with the increase of IPv6 technologies in web hosting and put it in perspective with network traffic increase in Japan; and offered a prefetching-based solution to increase cache hit rate and reduce response times on web browsers. [320] proposes to study DNS queries in their context by studying when DNS queries are performed and when the information is needed. Their conclusion regarding prefetching is that no supplementary DNS cost apply thanks to prefetching and that the overcost is minimal when data is not prefetched.

A good tutorial on prefetching and its consequences is provided by the chromium project ([321]).

We hypothesize that we can exploit DNS prefetching to query DNS servers based on devices mobility to resolve for device-specific information between the gateway and a DNS server. The prefetching can be as simple as requesting that nearby gateways prefetch the information but could also rely on recent mobility models based on ML predictions. The presented use case focuses on provisionning connectivity data based on the join exchange in the IoTRoam use case, but this method is applicable to other DNS data querying as defined at the beginning of this section.

This section studies the consequences of **prefetching DNS information on antennas** with regards to devices mobility but also studies **antennas occupation** based on mobility scenarios to further understand the possible impact of prefetching on antenna cache filling. We consider mobile vehicles in the Roma city, and we aim to analyze how we could reduce the overhead of DNS querying in our IoTRoam solution in a context of a vehicular application.

3.8.1 Use cases

Our IoTRoam use case 3.7 introduces two DNS queries for channel establishment between gateway and backend. Fetching data using DNS comes with a short delay. Our measurements from 4.10 showed that such a delay could be measured around 200ms. [234] studied DNS responses with overall results outlining this 200ms response for 70% of their queries, and 90% of queries are realized within 1s. More recent analysis, such as [255] or [256] outline better results by combining anycast technologies and Content Delivery Networks for DNS. [255] studies responses from top resolvers which answer 90% of their requests within 100ms. Moreover, [256] provides additional information regarding DoT resolution in which they outline failure rates with responses between 130ms and 230ms from top resolvers. Overall, the time inflation from additional security can be outlined around these values.

DoH would add another supplementary cost up to 150ms as outlined by [255] measurements on public resolvers. Adding an integrity check with DNSSEC would increase the requests even further. Overall, sending two complete DNS requests completed with integrity check and secured with DoH would cumulate up to 1.1s of queries done within the first exchange between the ED and the RG. Our problem is as such: "Would it be possible to reduce that delay in a mobility context to reduce the impact from DNS querying on channel establishment?"

This work provides a few insights on possible solutions based on Machine-Leaning-based mobility predictions and information prefetching from DNS servers.

We consider mobility traces from devices moving within the city of Roma; Figure 3.9 shows part of the studied traces traced as a function of latitude and longitude.

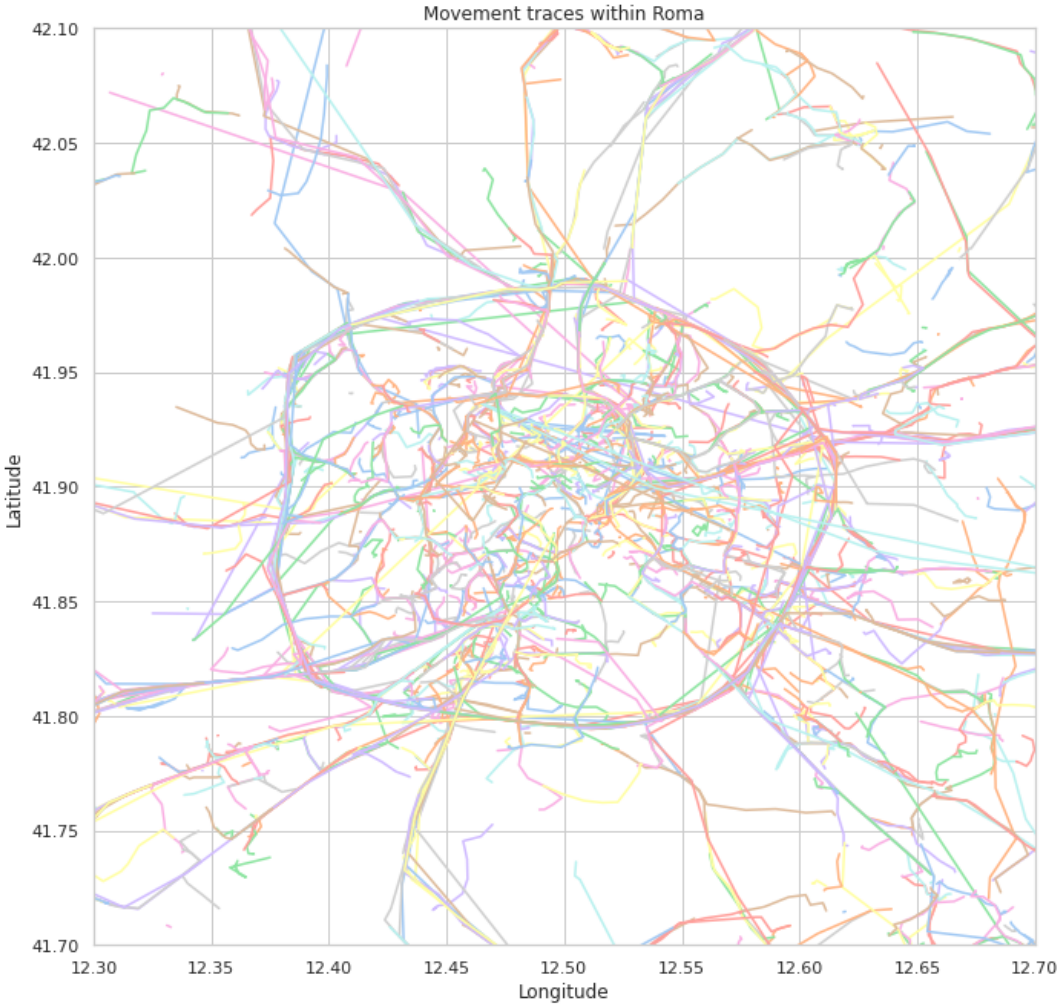


FIGURE 3.9: Vehicle mobility around Roma

We virtually place antenna within the movement perimeter. These antennas, regularly placed, provide independent coverage for our vehicles. Figure 3.10 shows a vehicular trace with the antenna disposition within its sector. Regularly placed antennas help us realize the closest one at glance, which is helpful when trying to infer the results and prototype. Our test antenna positioning algorithm places the antennas regularly in squares; thus, each antenna has 8 immediate neighbor for all scenarios.

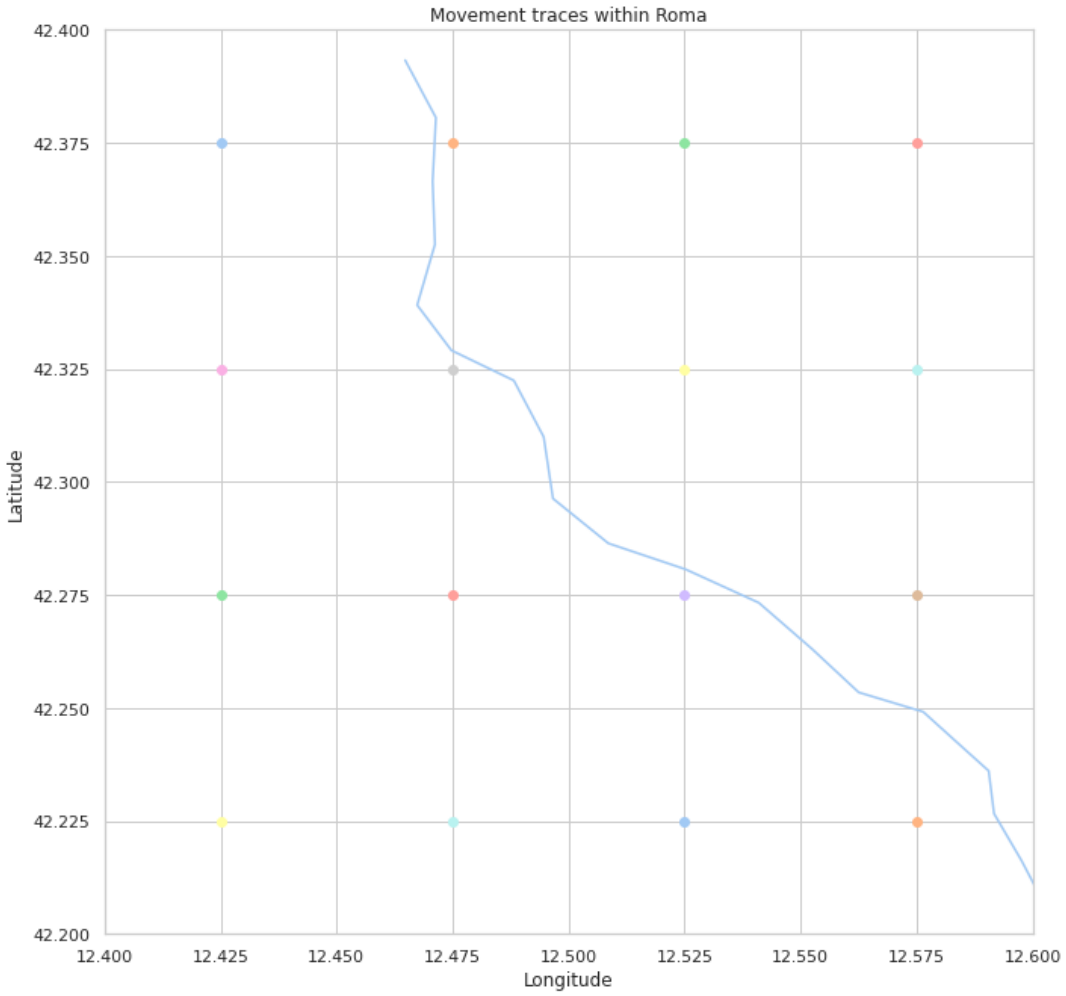


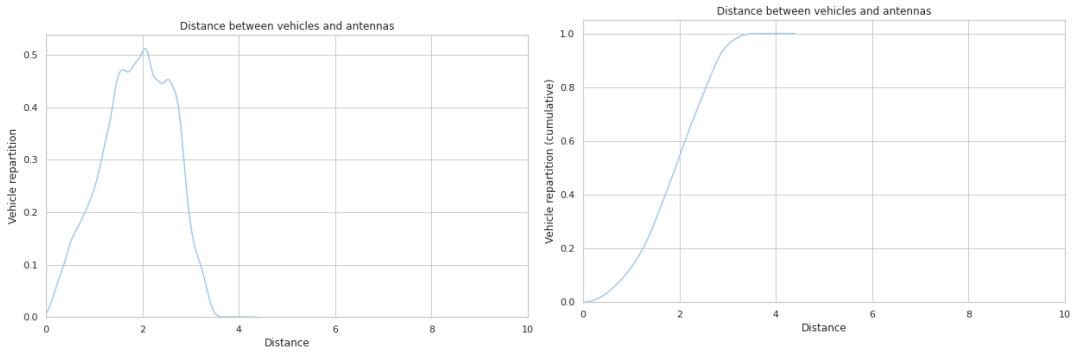
FIGURE 3.10: Vehicle and antennas around Roma

In our simulated fog LoRaWAN deployment, each antenna would act independently and provide access to its devices. To cover a city the size of our perimeter (200 km x 170km), a regular deployment will need around 520 independent antennas deployed.

Figure 3.11 provides insight on vehicle to antennas distances. With a regular antenna placement and about 8 km between two antennas at most, the vehicle-to-antenna distance will always be bounded between 0 and 4 km.

Based on our IoTRoam use case 3.6, we infer that each independent antenna will provide roaming access to devices within its reach. As described in the IoTRoam section, this means that the antenna will request the device's key from its HN and establish its connection to the ED thanks to them.

We separated our study into **three scenarios**. In the **first scenario**, **no prefetching** is realized, and the device uses the **standard DNS query** mechanism. In the **second scenario**, we improve the mechanism with a **basic prefetching** mechanism realized by **nearby antennas**. Finally, in the **third scenario**, we run **mobility predictions**, using ML, for our devices, plan their **possible future location** and **prefetch the information** based on the predictions.



(A) vehicle to closest antenna distance reparition

(B) Vehicle to closest antenna distance reparition (cumulative)

FIGURE 3.11: vehicle to closest antenna distance

3.8.2 First Scenario

For this first scenario, we studied the movements of 6992 devices within the Roma metropolis. Each vehicle is tied to 10 successive locations. We survey the nearest antenna for each location and check if the device's information is available on the antenna's cache or should be queried. Actually, depending on the vehicle movements, DNS configuration (number of entries in cache, TTL...) or antenna placement, it may come under the coverage of an antenna where it has already been before. The first location of the device is put on the side as "First DNS Query" for consistency with the other scenarios as we would not prefetch information for the first point of the time series.

Figure 3.12 describes our results. For the 10 locations of our 6992 vehicles, the antenna either query the DNS as part of the vehicle's first localization, query the DNS as part of an antenna change for the device or query its own cache as the device was already known.

Our studied traces are not heavily mobile for now as we study an urban scenario, and additional studies would be necessary to study possible other equilibrium between DNS caching and DNS querying for mobile devices.

Our first insight into these results would be that devices are moderately mobile, switching antennas once within the 10 points of their movements, moving around 35km per hour. We observe the 6992 initial DNS requests and around 8 thousand additional DNS queries, consistent with a 2.1 mean antenna per vehicle. The remaining DNS queries are prevented as the request hits the DNS cache within the antenna.

3.8.3 Second Scenario

For the second scenario, we studied the movements of the same 6992 devices; each vehicle is still tied to 10 successive locations. We survey the nearest antenna for each location and check if the device's information is available on the antenna's cache or should be queried.

Our test antenna positioning algorithm places the antennas regularly in squares; thus, each antenna has 8 immediate neighbors. In this scenario, we prefetch the information on these 8 closest antennas to anticipate possible device movements.

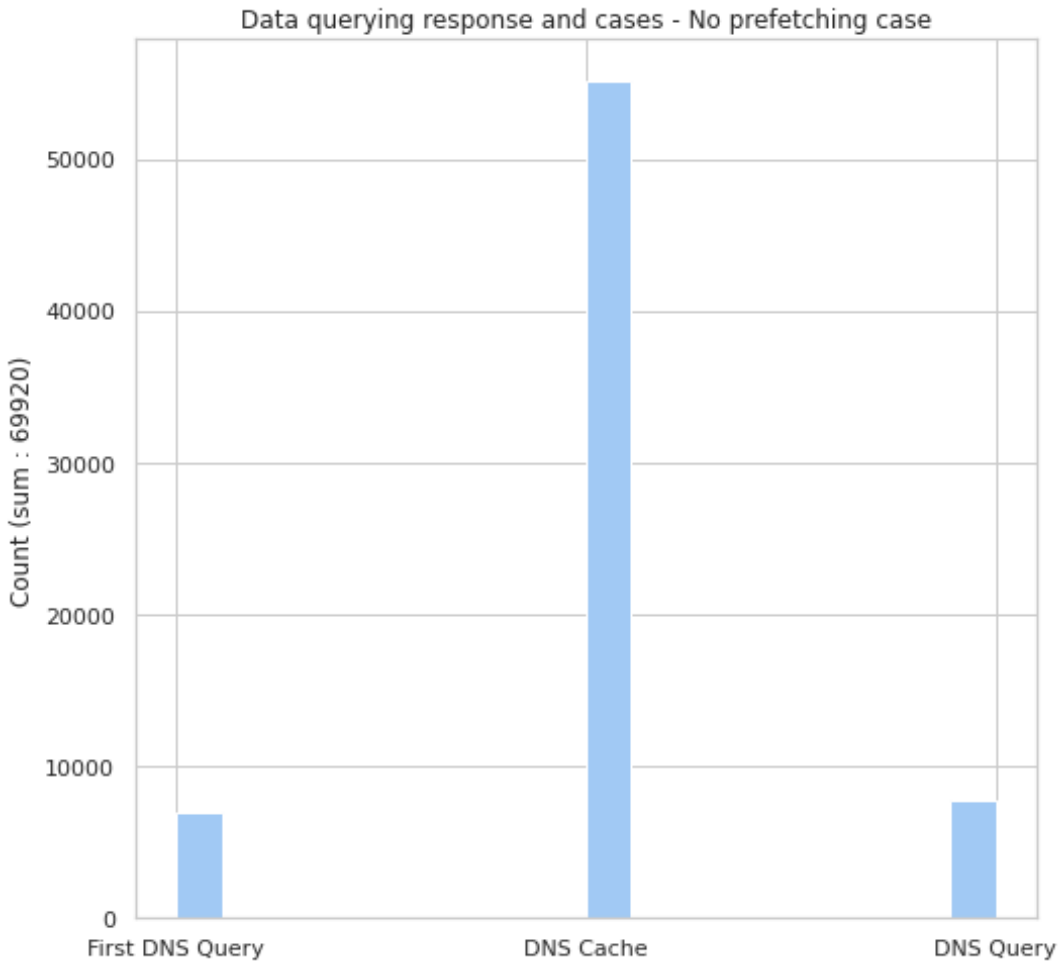


FIGURE 3.12: Cache Hit Rate distribution between queries - No Prefetching

As signalled above, the first DNS query for each vehicle is put on the side as "First DNS Query" as these DNS queries cannot be anticipated.

Figure 3.13 describes our results. As above, for the 10 locations of our 6992 vehicles, the antenna either query the DNS as part of the vehicle's first localization, query the DNS as part of an antenna change for the device or query its own cache as the device was already known through low mobility or prefetching.

The simulations show that prefetching permits us to prevent on-the-fly DNS querying. The DNS is still queried but at a time where the information is not yet necessary. The DNS cache handles all queries necessary for device communication, preventing additional DNS query time during handshakes. Nearby prefetching permits us to attain an important hit rate on our cache, whether filled by our first classic DNS query, DNS refreshes or prefetched DNS queries. A similar situation would be as described in the introduction of section 3.8, where prefetching every DNS zone encountered within web page URLs allow to quicken load time by pre-filling the DNS cache with prefetched DNS queries.

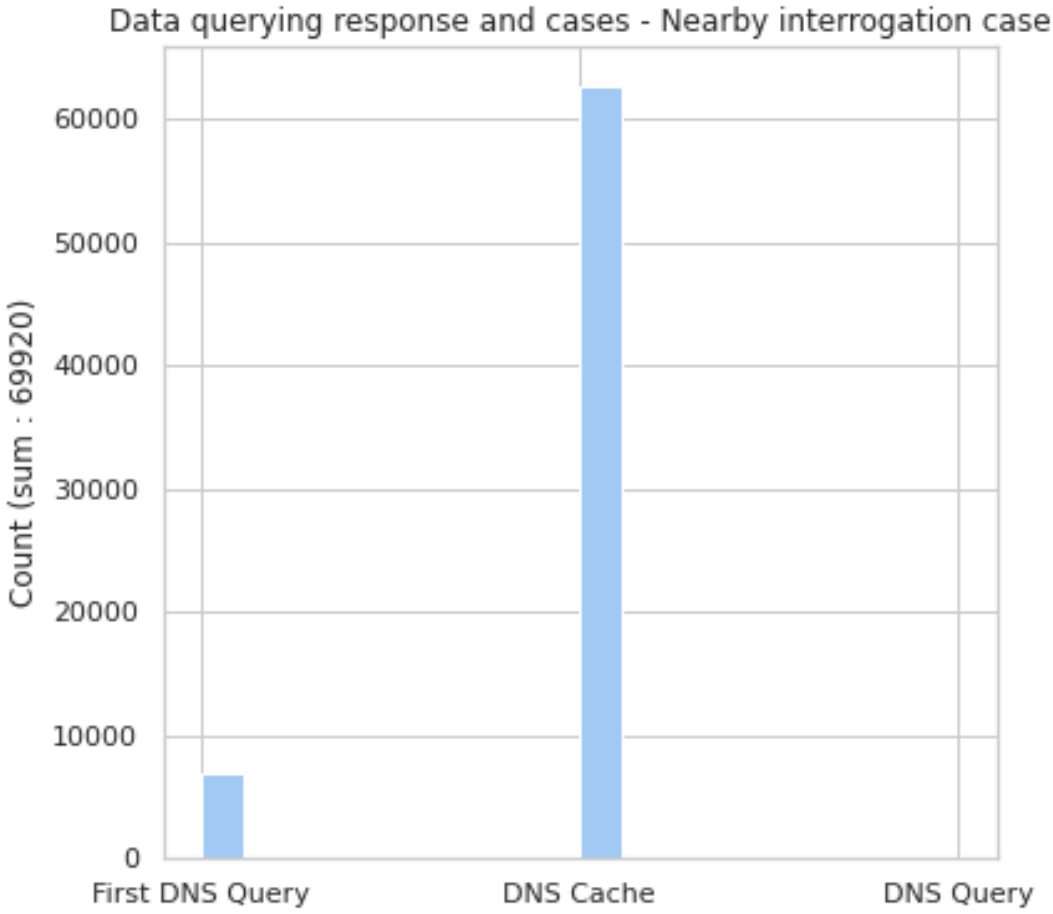


FIGURE 3.13: Cache Hit Rate distribution between queries - Nearby prefetching case

3.8.4 Third Scenario

In this third scenario, we predict car mobility using deep learning algorithms and identify antennas candidate for device coverage. Based on these predictions, the DNS (or its cache) is queried once by the antenna corresponding to the device's position for each given point within the device's movement. Then for the four following predicted positions, the corresponding antenna will perform DNS prefetching to heat its cache for a possible change of coverage from the device.

Figure 3.14 provides a rundown on interactions between antennas and DNS Servers in the third use case.

For a given position A, we consider the 25 possible antennas (B to Z) from the previous prediction and positions of the vehicle:

- Antennas B to E are antennas corresponding to the prediction of position A in previous moments in time $\{f_{T-i}(T+i), i \in [[1,4]]\}$). If antenna A corresponds to one of these antennas, we consider that our prediction is successful, and we hit the cache of our gateway as the information was prefetched in previous moments in time.
- Antennas F to O correspond to the predictions for previous positions of the device ($\{f_{T-i}(T+j), i \in [[1,5]], j \in [[1,4]], i-j>0\}$). If antenna A corresponds

to one of these antennas, but not antennas B to E, our prediction was a failure, but the actual corresponding prediction was correct with a time shift. Furthermore, the prefetching for these antennas was realized; thus, the information is still present in the gateway's cache, and despite the prediction failure for this exact timestamp, we hit our gateway's cache as the information was not purged yet.

- Antennas P to S are a similar case ($\{f_{T-i}(T+j), i \in [[1,5]], j \in [[1,4]], i-j < 0\}$), our prediction was a failure, but the predicted antennas was correct considering a time shift (and would probably be correct for future device positions). Furthermore, the prefetching for these antennas was realized; thus, the information is present in the gateway's cache, and despite the prediction failure for this exact timestamp, we hit our gateway's cache as the information was not purged yet.
- Finally, antennas V to Z are the actual antennas solicited for the device in previous moments in time ($\{f_{T-i}(T), i \in [[1,5]]\}$). Should all other prediction fail but antenna A correspond any antennas from V to Z, the prediction is a failure and so is the prefetching as the other prefetched information expired, but the information corresponding to these antennas are still present in the DNS cache from previous requests, we labelled this result "DNS Cache - No Prefetch"
- In the case where antenna A ($\{f_T(T)\}$) does not correspond to any antenna between B and Z, prefetching was a failure, and a new antenna was solicited; thus, it must realize a DNS request (labelled "DNS Query")
- Additionally, we separated from these DNS queries the DNS query for the first device's location as antenna B to Z constitute an empty ensemble for this given location.

	Actual position	T+1 Prediction	T+2 Prediction	T+3 Prediction	T+4 Prediction
Antenna ID (T-5)	Z	L	M	N	O
Antenna ID (T-4)	Y	I	J	K	E
Antenna ID (T-3)	X	G	H	D	S
Antenna ID (T-2)	W	F	C	Q	T
Antenna ID (T-1)	V	B	P	R	U
Antenna ID (T)	A				

FIGURE 3.14: Possible solicited antennas in Scenario 3

Figure 3.15 combines the results from the 69920 vehicle location based on the scenario breakdown from figure 3.14.

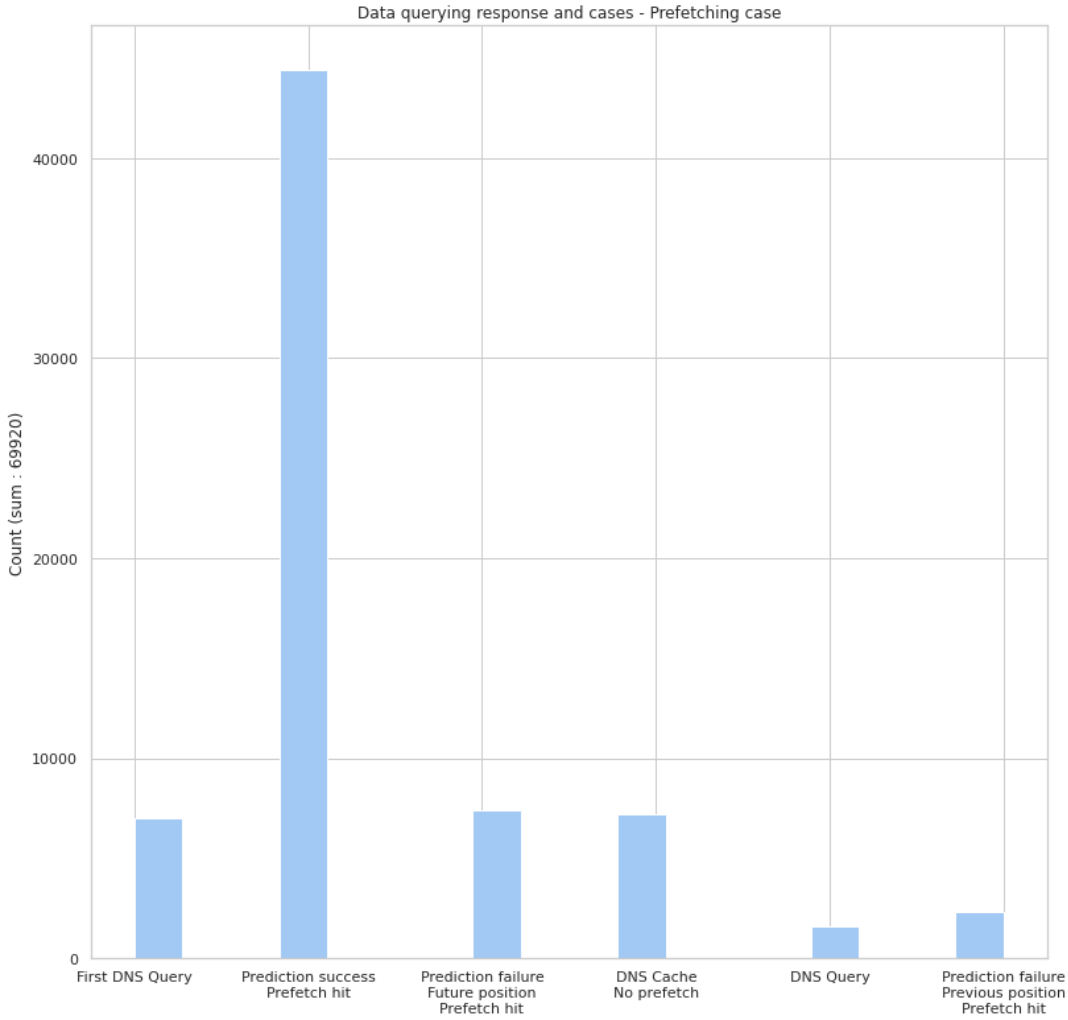


FIGURE 3.15: Cache Hit Rate distribution between queries - Predictor prefetching case

Results are, satisfying compared to the first scenario. Successful prediction lead to hitting an antenna linked to a correctly predicted position in 63.4% of cases. Cache hit rate linked to predictions, whether correct predictions or incorrect prediction by lateness or earliness, would add up to 77.4% of requests. The remaining 22.6% are divided between the first query (10%), DNS cache after prediction error (10.3%) and actual DNS queries (2.3%).

3.8.5 Antenna occupation

Another important subject to study is antenna occupation. As part of our study, antennas prefetch information based on the possibility that the associated device will pass under its coverage. What is the consequences of such additional operation on our antennas:

- We placed **520** virtual antennas around the city
- Out of them, the first scenario activates **301** antennas. That means that our 6992 vehicles pass near these 301 antennas and that 301 is our minimum number of active antennas as a whole.

- The second scenario activates as whole **393** antennas, a bit over twice more antennas than in the first scenario. The 'nearby case' shows excellent results but would probably create congestion within the network should these results confirm at a larger scale.
- Finally, the third scenario activates **380** antennas, globally around the same amount as the antennas solicited as part of the second scenario, figure 3.16 gives us more insight on the distribution of these antennas.

Figure 3.16 shows the comparison of the number of activated antennas between scenario 2 and scenario 3.

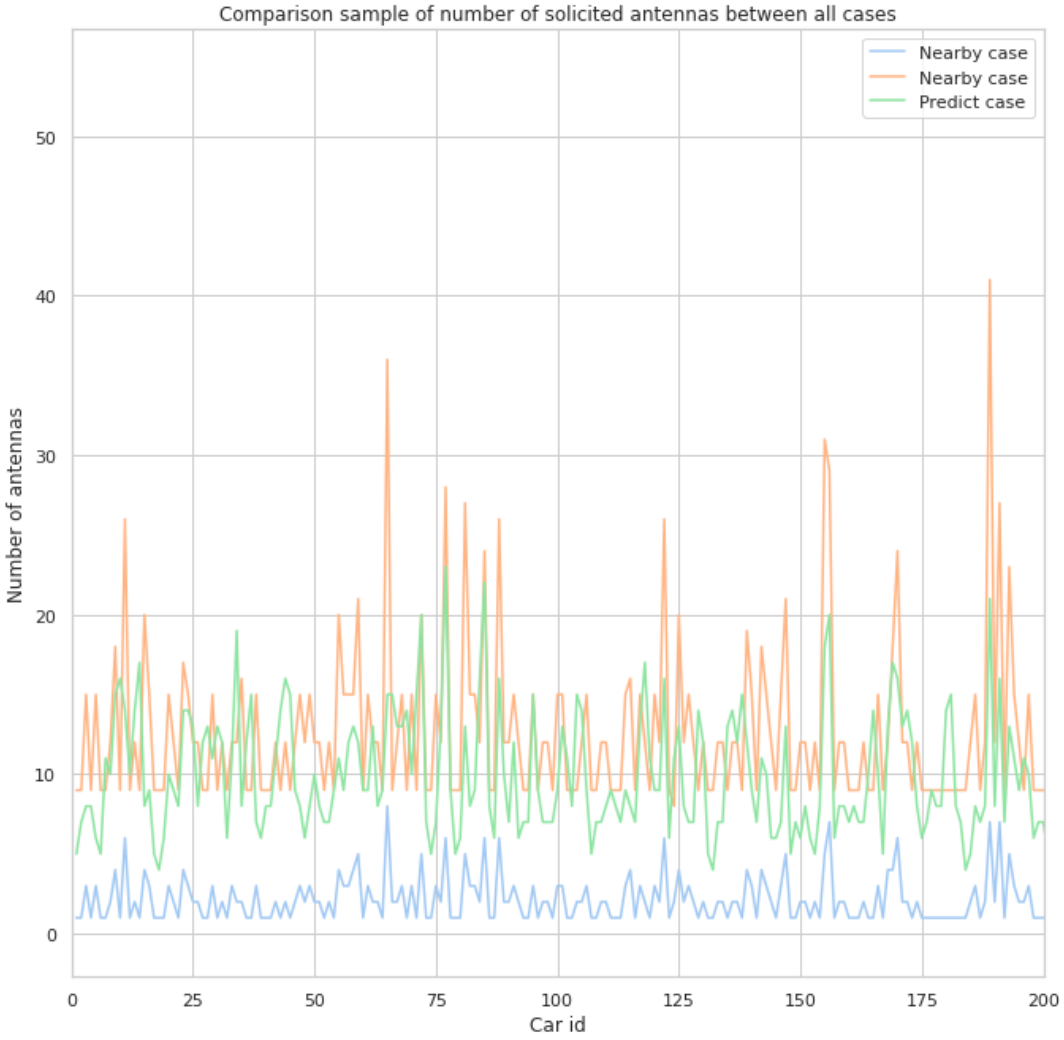


FIGURE 3.16: Sample from activated antennas in all scenarios

The mean amount of antennas, described on 3.17 activated is as follows:

- Scenario 1 leads to activating 2.1 antennas per moving vehicle on average.
- Scenario 2 leads to activating 12.3 antennas per moving vehicle on average.
- Scenario 3 leads to activating 9.7 antennas per moving vehicle on average.

Figure 3.17 provides additional insight on these values, Scenario 1 has at least 50% of its values between 1 and 3, Scenario 2 between 9 and 14 and Scenario 3 between

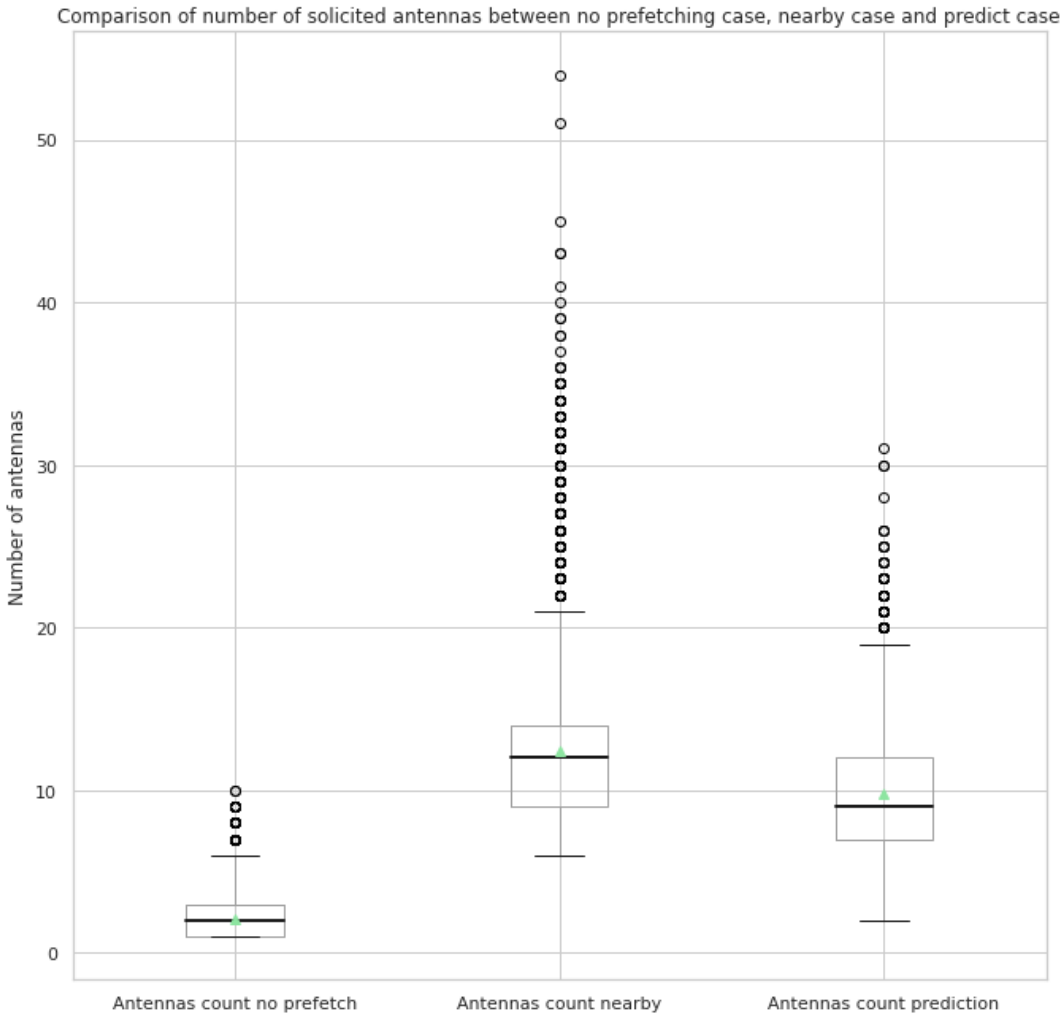


FIGURE 3.17: Activated antennas repartition for each scenario

7 and 12. This result fits with the moderate mobility from our values as devices that move within 3 antennas would activate within to 15 through their movement in Scenario 2. The predictor performs better than the simple nearby prefetching, with more than 75% of its values under the median for Scenario 2. Also, Scenario 2 has many outliers with over 21 antennas solicited per device on highly mobile roads, in which the predictor performs better.

3.8.6 On prefetching efficiency

DNS prefetching is an **efficient tool** to **reduce on-the-fly DNS queries** necessary for devices communication. Prefetching the information on nearby antennas can completely prevent DNS queries by performing them in advance around the closest antennas, but at a cost as devices request more antennas, especially in a highly mobile environment.

By exploiting recent ML capabilities for traffic prediction, we could provide a solution that **heats the cache** for 78% of requests and that lead to a cache hit for 88% of them, the remaining 12% consist of the first DNS query (10%), and on-the-fly DNS queries following prediction failures (2%).

Overall, the ML system would outperform its nearby-activation counterpart in terms of antennas solicitation. However, additional simulations with different antenna positioning would be interesting to study. Another interesting point would be the study of possible antennas overload. Also, considering that our traces amount for taxis which represent around 1% of actual cars circulating around a country, studying actual overload within the network by increasing the number of vehicles by a 100-factor then decreasing it considering the number of cars that would actually transit within the system would be feasible.

3.9 Contributions

The IoTRoam experience enabled us to set up a federated platform that has been documented and the software provided as open-source to the community. It also helped us to identify operational issues that have not been encountered earlier since there is no LoRaWAN operational infrastructure using DNS for OTAA and Roaming. The PoC tests proposed solutions to some of the operational issues and also led to **three change requests** adopted by the LoRaWAN backend specifications. This section will detail the contributions.

3.9.1 Contribution 1

The networks based on 'One-to-One' interconnection or hub drop the incoming packet from an ED if it is not part of its network or its partners. With the IoTRoam federated model, these networks could make a DNS resolution to identify the HN of the ED. Thus, the IoTRoam federated model caters to the whole ecosystem wherein networks based on the 'Hub' or 'One-to-One' interconnection could co-exist.

3.9.2 Contribution 2

In the cellular model, portability between operators becomes possible since there is a human subscriber involved, which is not the case in LoRaWAN. In LoRaWAN, the EDs with a battery life spanning for a decade are supposed to be set up in remote places and not readily accessible in the necessity of a network operator change. IoTRoam enables **portability** between different operators; thanks to the DNS database, the JS pointing to a JoinEUI can be modified without making any modification at the ED level.

To understand the importance of operator portability, a brief background of how the ED is provisioned with the JoinEUI and DevEUI are needed. The JoinEUI 64 bit address could be divided into three broad ranges: **OUI** of the manufacturer, the **Batch ID** of the manufacturer and the **JoinEUI** value assigned to the batch. The DevEUI is also a unique IEEE EUI-64 bit address divided in the same categorization as the JoinEUI. The difference is - for every ED there is a unique DevEUI, but thousands of EDs could be assigned a single JoinEUI as shown in the table 3.1:

During the JoinEUI assigning process, the ED manufacturer is not yet aware of who will be the buyer. If a client is buying only 500 EDs from a batch of 1000, the remaining 500 EDs' JoinEUI need to be re-provisioned with a new JoinEUI if a new buyer wants the remaining 500 EDs to point to a different JS. Similarly, if the buyer who has bought 500 EDs from a batch needs to assign a different JS for a set of 100 EDs, the JoinEUI needs to be modified in each ED. This modification is done by re-flashing the EDs with the new JoinEUI and thus is operationally time-consuming and costly.

TABLE 3.1: Fictional representation of how DevEUI and JoinEUI 64 bits are partitioned, wherein certain bit blocks are allocated for OUI, certain bits for the batch (e.g. ABBB) & the remaining bits at the serial level

DevEUI	JoinEUI
OUI-ABBB-0001	OUI-ABBB-FFF1
OUI-ABBB-0002	OUI-ABBB-FFF1
OUI-ABBB-0003	OUI-ABBB-FFF1
....	...
OUI-BBBB-0001	OUI-BBBB-FFF2
OUI-BBBB-0002	OUI-BBBB-FFF2
OUI-BBBB-0003	OUI-BBBB-FFF2
....	...

The PoC experience enabled us to suggest a **change request** to provide an operationally feasible solution, which has been accepted and included in the LoRaWAN backend specifications. The solution proposed by the change request is to create a combination of the DevEUI (which is unique for each device) and JoinEUI and provision them in the DNS. In order to adapt to this requirement, the NS should first make a DNS query using the concatenation of DevEUI and JoinEUI, and if the resolution fails, it falls back in making a DNS query only using the JoinEUI.

Taking an example where two EDs (**0xACDE480001020234**, **0xACDE480001020ABC**) should point to two different JSs, but has a single JoinEUI represented in the hexadecimal format as **0x00005E100000002F**. The DevEUI JoinEUI combination could be provisioned in the DNS pointing to two different JSs as follows:

4.3.2.0.2.0.1.0.0.0.8.4.e.d.c.a.f.2.0.0.0.0.0.0.0.1.e.5.0.0.0.0.
joineuis.lorawan.net IN 192.168.2.4
a.b.c.0.2.0.1.0.0.0.8.4.e.d.c.a.f.2.0.0.0.0.0.0.0.1.e.5.0.0.0.0.
joineuis.lorawan.net IN 192.168.2.5

Based on the longest match algorithm, the DNS resolution will resolve to two different JS's for the two ED's even though the JoinEUI are the same for both the EDs.

3.9.3 Contribution 3

The second **change request** that was adopted into the LoRaWAN backend specification includes modifying the subdomains for join and roaming from **lora-alliance.org** to **lorawan.net**, thus separating the LoRa Web and DNS service.

3.9.4 Contribution 4

A third **change request** that was adopted into the LoRaWAN backend specification consists of updating the DNS provisioning and resolution section to enable the usage of any DNS resource record for OTAA and roaming functionalities. Before the change request, the LoRaWAN backend specifications were normalized using a NAPTR DNS resource record, which is considered quite complex (Explained in RFC 3401, 3402 & 3403) for operational purposes.

3.9.5 Contribution 5

We developed and provided a secure, automatized DNS provisioning platform that could be used by the community. With a secured API key, any authorized user can access the User Interface (UI) (via web or API). The UI enables authorized users to do multiple operations (creation, modification, deletion) of only their data in the DNS database.

While testing the UI with some LoRa Alliance community members, we encountered operational issues such as validating that the data provisioned in the DNS is done by the rightful owner. The need to validate the JoinEUI (an IEEE EUI-64 identifier provisioned by the IEEE and has OUI in the IEEE EUI-64) with the IEEE OUI database, were identified and implemented, thanks to the PoC. The implemented solution has been provided as feedback to the LoRa Alliance, which could be integrated when the DNS service operated by the LoRa Alliance is deployed.

3.10 Conclusion

Our objective with IoTRoam is to achieve the same service as cellular or Wi-Fi roaming built on a global resolution and security infrastructure, namely the DNS and PKI. We added a hard requirement that the infrastructure or technologies used to achieve this vision should be viable, operationally feasible and could be integrated into existing IoT infrastructures with minimum changes. To achieve our vision, we followed the WBA guidelines for open roaming and satisfied the requirements outlined by employing open standards used on the Internet, such as DNS and PKI.

We chose LoRaWAN, an evolving standard, and demonstrated that seamless IoT roaming with minimum prior configuration is possible using the IoTRoam architecture. In this process, we have deployed a PoC and provided all necessary building blocks (documentation, software, UI, video tutorial) so that each one in the community could make use of them to implement his own network.

This experience has also helped us to propose three Change Requests that have been adopted into the LoRaWAN Backend Interface Specification. The first one includes the possibility of using any DNS resource record ED activation and roaming functionalities. The second is creating a combination of the DevEUI (which is unique for each ED) and JoinEUI and provision them in the DNS. This solution was proposed to resolve the device manufacturer's issue of providing the ED's configured in the same batch with same the JoinEUI and different DevEUI to be sold to different buyers. The third includes modifying the domain names for join and roaming from *lora-alliance.org* to *lorawan.net*, thus segregating the LoRa Alliance Web and DNS service.

The IoTRoam initiative was advertized through various press releases and with academic partners. We discussed running interoperable testing using the federated platform with several institutions in France, Denmark, and Italy. We also discussed with network operators to run experiments with massive public network infrastructures. Running additional tests with these institutions would help us study the impact of heterogeneous backend infrastructures and their effect on the quality of the communication channel. It would also allow us to gather additional data on the impact of DNS complete resolution on the LoRaWAN/IoT traffic.

As the objective is to interconnect networks using different IoT technologies, the next steps consist of testing roaming interoperability with NB-IoT, 5G or Wi-Fi. For ED onboarding, we are also working on integrating DANE with DNSSEC since the certificate data itself can be stored in the DNS, possibly replacing or completing the PKI.

On the subject of reducing the impact from DNS, we studied through simulations the consequences of caching and prefetching DNS information with mobile devices in a city. Our combination of an ML predictor and prefetching allow for an interesting reduction of DNS requests realized compared to a caching-only solution and a reduction in the number of gateways realizing prefetching operation compared to soliciting the closest nearby antennas. This mechanism, already crucial for cloud providers in the context of the development of the web, proves efficient and useful for operators and asks interesting question on locating caches and optimizing caching delays.

The application of DNS prefetching in a mobility context is applicable when querying other connectivity information from the DNS infrastructure such as the querying presented in Chapter 4 or the one presented in Chapter 5, Section 5.5.

Chapter 4

DNS-based dynamic context resolution for SCHC

4.1 SCHC, connecting LPWANs to the IP stack

Complementary to developing LPWANs infrastructure, a key aspect in developing LPWANs relies on connecting them to the Internet. Indeed, LPWANs are fundamentally non-IP networks. We already detailed a few constraints of LPWAN communications in the previous chapters. The maximal frame size for LPWAN payloads, a constraint that limits the connection of LPWANs to the Internet drastically. Table 4.1 explicit this limitation. When working with small frame sizes such as LoRaWAN's or SigFox's, signalization size, and in the case that interests us, headers size, become a significant issue. Table 4.2 sums up the main header size for layer 2, layer 3 and layer 4 when working with various technologies usually embedded onto wireless devices, as well as the percentage of frame size it corresponds to in the context of LPWANs.

Based on this observation, the lpwan working group designed a framework to reduce the IPv6 header size to embark the IP stack onto LPWAN devices. **SCHC** [3] is a framework that provides both compression and fragmentation functionalities. It was standardized by the lpwan [161] working group at the IETF. It is considered an efficient solution to **connect the LPWANs to the Internet using IPv6**, thus enabling end-to-end IP connectivity. Figure 4.1 illustrates SCHC compressing capabilities and effect onto layers with regards to each layer size in different connectivity setups. With the help of the SCHC framework, it is possible to compress an IPv6 header from its original size of sixty bytes down to two bytes, thus reducing bandwidth usage and increasing communication efficiency. Enabling IPv6 connectivity for LPWANs is a key issue to connect the LPWANs to the Internet via the IP stack.

The SCHC solution was also designed to break the IoT siloes. Using SCHC allows applications embedded onto devices to communicate using a common, standard system, the IP stack, with SCHC handling the adaptation between the applicative and lower layers, specific to each IoT technology.

	LoRaWAN (bytes)	NB-IoT/LTE-M (bytes)	SigFox (bytes)
Frame size	250	1600	29

TABLE 4.1: Max Frame size from the main LPWANs technologies

Headers	LoRaWAN	NB-IoT/LTE-M	SigFox
L2 header	8 octets 3.2 %	14 octets 0.875 %	10 octets 34,4 %
L3 / IPv6 header (40 bytes)	16 %	2.5 %	138 %
L4 / UDP header (8 bytes)	3.2 %	.5 %	27.6 %
L5 / CoAP header (4 bytes)	1.6 %	.25 %	13.8 %
L3+L4+L5 / SCHC (2 bytes)	0.8 %	.125 %	6.9 %
Cumulative (no SCHC)	24 %	4.125 %	213.8%
Cumulative (SCHC)	4 %	1 %	41.3 %

TABLE 4.2: Frame Header Occupationas percentage of frame size for the main LPWANs technologies

FIGURE 4.1: SCHC compression capabilities (Source [322])

Finally, SCHC enables technologies to be embedded with a common, global, underlying adaptative technology, assisting developers when working on IoT solutions allowing them to work regardless of the actual underlying layers.

SCHC is designed keeping in mind that the LPWANs are star-oriented technologies, with EDs usually communicating to a single network gateway or linked to a single backend element. Thus its operating principle is based on the assumption that the device will compress data, and the center of the star topology, usually the network gateway or another centralized backend element, will take care of the decompression. The framework also considers that the IP header expected for LPWANs EDs can be predefined as the ED is less likely to change its internal software and usually evolve in a known network environment. To compress the data sent and received between the ED, its backend element, SCHC uses a predefined group of rules called **context** which is deployed on the ED and one of the backend element (RG or backend servers). This context may be specific for each ED or common for a group of EDs. Fig. 4.2 presents such an example of such a context rule.

Based on this context rule, a pattern matching operation is realized on the packet header emitted by the LPWAN ED. These operations usually **match each IP header field** with their expected values and remove them if the values correspond to the

Rule 0

Field	FL	FP	DI	Value	Match Opera.	Comp Decomp Action	Sent [bits]
IPv6 Version	4	1	Bi	6	ignore	not-sent	
IPv6 DiffServ	8	1	Bi	0	equal	not-sent	
IPv6 Flow Label	20	1	Bi	0	equal	not-sent	
IPv6 Length	16	1	Bi		ignore	compute-*	
IPv6 Next Header	8	1	Bi	17	equal	not-sent	
IPv6 Hop Limit	8	1	Bi	255	ignore	not-sent	
IPv6 DevPrefix	64	1	Bi	FE80::/64	equal	not-sent	
IPv6 DevIID	64	1	Bi		ignore	DevIID	
IPv6 AppPrefix	64	1	Bi	FE80::/64	equal	not-sent	
IPv6 AppIID	64	1	Bi	::1	equal	not-sent	

FIGURE 4.2: Context rule example as presented in the RFC 8724 (Source [3])

rule defined in the context. When an ED receives a SCHC compressed packet, the reverse operation is realized to **build the IP header back**, allowing the applications to communicate on the network without considering the underlying IoT specificities.

To prove the operational feasibility of SCHC, members of the lpwan working group at the IETF designed a PoC implementation called OpenSCHC ([323]). OpenSCHC is a reference Python codebase for the SCHC protocol, which we used in our experiments. SCHC works as follow: when sending data from the ED to its RG, the SCHC context rules enable compression by **suppressing redundant**, **superficial**, **predictable** or **most used data** inside an IPv6 header and replacing them with a **Rule Identifier** (Rule ID) chosen in a given set of predefined rules. Among multiple rules hosted on the devices, a single rule is selected to fit as precisely as possible to the corresponding application that needs to transmit its data. All the rules associated with a given ED are hosted on its corresponding backend to realize the opposite operation and decompress the data received over the air.

Our interrogation regarding SCHC can be summarized as follow: SCHC is a protocol built on static information. It relies on identical information stored on both the ED and the network backend. This identical information, the context, usually consists of multiple rules corresponding to the associated ED. When using SCHC, one element from the backend is supposed to realize SCHC operation (compression, decompression, fragmentation, reassembly) for all associated EDs. Allowing the owner to host his rules and to modify them quickly at a later date, storing only a piece of information on either the RG, the NS or the AS such as the Rule Identifier or Version might help to introduce more flexibility to the SCHC protocol and simplify the user's maintenance. Also, storing all SCHC rules, considering they might be unique for each ED, might introduce scalability issues to the system. We can consider around 20 rules per ED when working with such rules, with thousands of EDs around a single antenna and multiple antennae for a given server. LoRaWAN, built as a star of star topology, is a good example where multiple RGs, each handling multiple EDs, can be connected to a single NS. When considering hundreds of thousands of EDs around a single server with up to 5 kb per context rule, we end up storing **gigabytes** of data to enable SCHC on a given LoRaWAN infrastructure. Finally, considering SCHC's static design, building a system supporting both SCHC and roaming might prove

complicated. SCHC's static design might harm communication when working in a mobility scope, typically when an ED is roaming.

To assist SCHC's development in these three scopes: **hosting rules**, **supporting scalability** and **developing mobility scenarios**, we propose that the RG, NS or AS retrieve the context **dynamically from a remote server**. Hosting such information on a remote server separates traffic from its underlying protocol, separating the interfaces, each dedicated to its use. Storing outside the main servers' scope lightens the weight associated with local storage by introducing a tradeoff between storing SCHC's most used rules and querying rules used less often. Retrieving the context from an accessible remote server strengthens mobility and roaming capabilities for EDs, which can work anywhere once their context is retrieved.

This section provided information regarding SCHC, such as the motivation that leads to its development and the questions that we aim to address in this chapter. We presented the SCHC framework and explained how, by enabling IPv6 for LPWAN communication, SCHC creates a bridge between the siloed LPWAN and the Internet. In section 4.2 we present the experiment we realized as part of our work with SCHC, the key research issue we identified, the mechanism we designed to address it and the experimental setup we deployed to test our hypothesis. Lastly, we present our experimental results in 4.3, regarding the SCHC framework and how our mechanism introduces more flexibility in the system. Results that we discuss further in section 4.4 by opening the discussion on possible evolutions to the framework.

4.2 Experimenting with SCHC and DNS

There are multiple options for storing these context rules. It could be done in a private server, using, for example, an Administration Management System as proposed by [4], stored in the cloud or even shared on a blockchain. However, we hypothesize it could be wise to use an open, distributed mechanism to find the location of the server where the context rules are stored. We propose to experiment possible use of **DNS as a way to support SCHC compression**. As an optimized, hierarchical and distributed database, DNS could support the determination of the location of the server where the context rules are stored feasibly on the Internet. Hopefully, using such a mechanism would allow for a seamless transition, from preconfiguring the information needed on the backend to building it dynamically, on the fly, based on actual needs when operating the network.

DNS would prove an efficient solution to introduce more flexibility and improve scalability when using SCHC. Our solution aims to provide open access to SCHC parameters to support **roaming capabilities**, **improving flexibility** and **scalability**. We study the possibility of introducing a context registry outside the scope of the ED's NS. To study this problem, we deployed a dynamic context resolution architecture based on DNS for SCHC compression and decompression and studied its consequences on system latency and LoRaWAN communications.

Our experiment proposal for this section aims to study and improve SCHC's compression and decompression mechanisms. It relies on multiple scenarios, defined in the following parts of this section, that aim to study SCHC compression and decompression mechanism and the delays added by the system concerning transmission delays. Then we added two possible remote rule management systems to introduce more flexibility to the SCHC standard to solve the scalability issue identified in the

previous subsection and permit SCHC rules to be easily shared across the network, for example, in a roaming context.

Fig. 4.2 presents such a context rule whereas the experimental testbed and its components is described in 4.2.3.

In order to prevent synchronization issues, a frequent issue when working with constrained devices such as LPWAN-enabled devices, the delays are always measured on either the ED (e.g. full round trip time) or the backend (e.g. DNS resolution)

For our experiments, we chose not to focus on transmitting the rules to the devices over the air and considered that the rules would already be on the devices as the current standard proposes it.

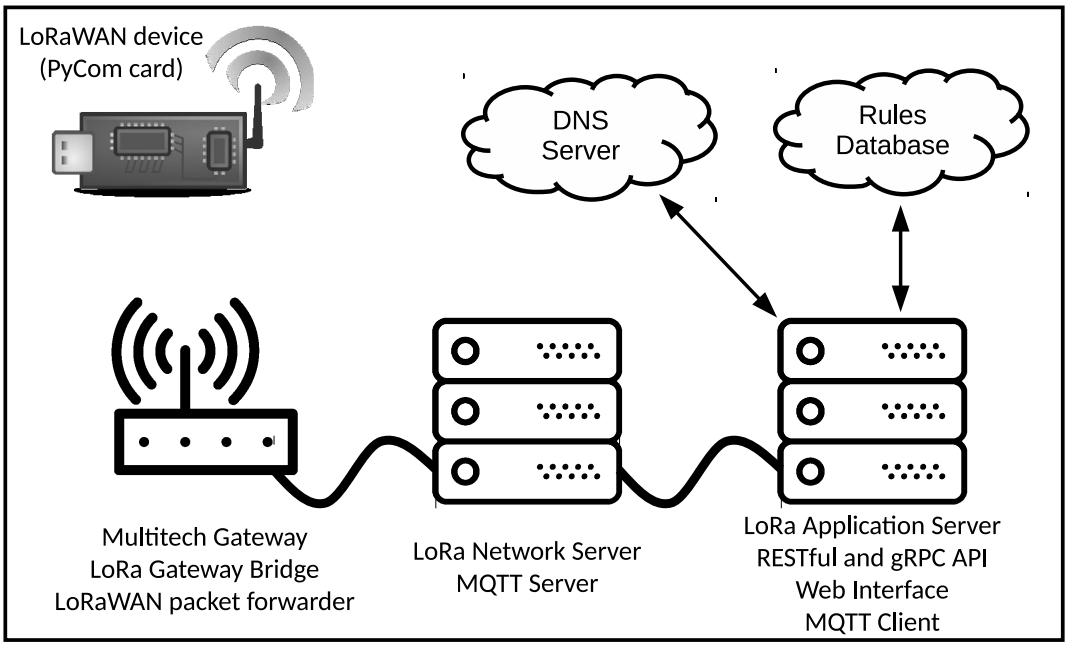


FIGURE 4.3: Measurement Platform's Network and system architecture (rework this scheme)

4.2.1 Proposing querying mechanisms for context resolution

Querying information can rely on various technologies: Edge technologies ([48], [49]) can be leveraged to place information in the most efficient location possible wih regards to the ED's location. [56] propose a routing method to query information more efficiently and [57] proposes a data-centric approach intended for low power networks. Another key evolution in edge technologies is the Information Centric Network ([324], [325]) approach that rely on a decentralized naming convention to distribute the information efficiently within the network. Our method differs as we leverage known techniques from the DNS world to improve Context rules distribution.

The proposed mechanism delocalizes the SCHC rules on a **remote server** and uses **DNS** to retrieve them. When the ED sends data, they are received by the RG, then transferred to the NS and retrieved by the AS. To decompress the data, the AS needs access to the rule. We use the DNS to retrieve a hash of the rule (since it is not possible to store the entire rule in the DNS) and possibly an address of an HTTP

server on which the corresponding rule might be stored. An HTTP server is used to store the rules themselves. The rules can be uniquely linked to the tuple **(DevEUI, RuleID)**. This tuple is constructed by extracting the DevEUI from the LoRa frame and the RuleID using the first bits from the LoRa payload compressed by SCHC. The AS stores the rules corresponding to the device under its coverage in a rules cache for a set duration. The rules in the AS are indexed in a hash table. When the AS receives data, it constructs the tuple (DevEUI, RuleID) as indicated above, then uses the DNS to retrieve the hash of the corresponding rule and search for this rule in its rules cache. If it is not found because it is a new tuple (DevEUI, RuleID), a new rule must be stored in the **cache**, and the HTTP server where the rule is stored is interrogated to get it. Then, the rule is inserted into the cache. Note that even when a rule is present in the cache, the DNS is systematically queried because the **freshness of the information** must be checked to ensure that the rule has not been modified since its **last cache insertion**.

Finally, the data can be decompressed. Once the data are decompressed, the server may send a response back depending on the application's needs. Fig. 4.3 presents the interactions between the AS, the DNS and the HTTP server in the case where a new rule is needed.

4.2.2 Measurement scenarios

Our study focuses on **AS Response Time** and **ED Uplink Round Trip Time** (U-RTT) through different scenarios. Scenarios 1 and 2 serve as references to compare with other scenarios. They provide the minimum communication time with and without SCHC decompression. Scenario 3 aims to study the mechanism presented in 4.2. Scenario 4 studies the case where most of the information is always present in the cache. These four scenarios are described more precisely below:

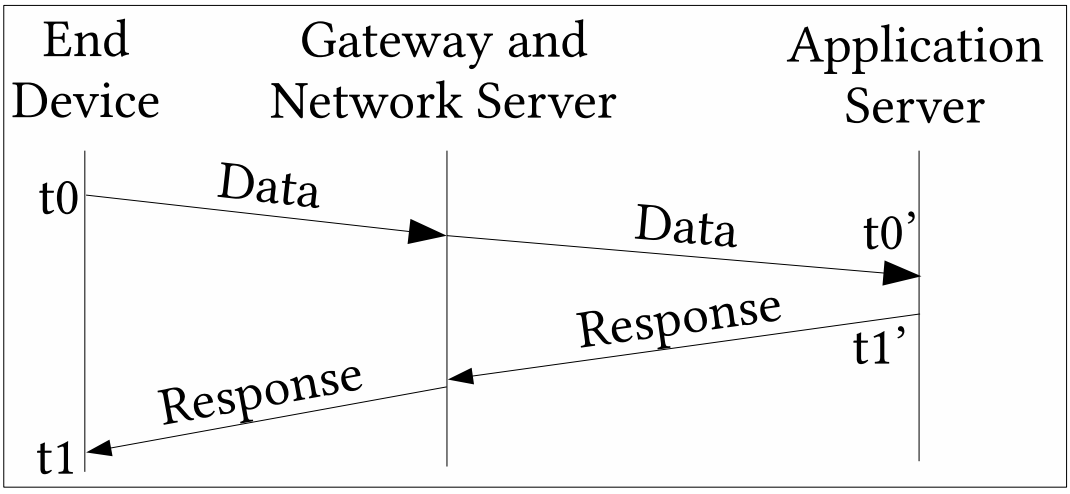


FIGURE 4.4: Message Exchange Diagram (Scenario 1)

- Scenario 1: The first measurement is designed to be used as an **experimental reference** for our platform. Data are sent **without compression** from the ED over LoRa and a response is sent back from the ChirpStack AS in order to measure the RTT $t1 - t0$ (cf. Fig. 4.4). We also measure the AS Response Time $t1' - t0'$. No decompression operation is performed on the data.

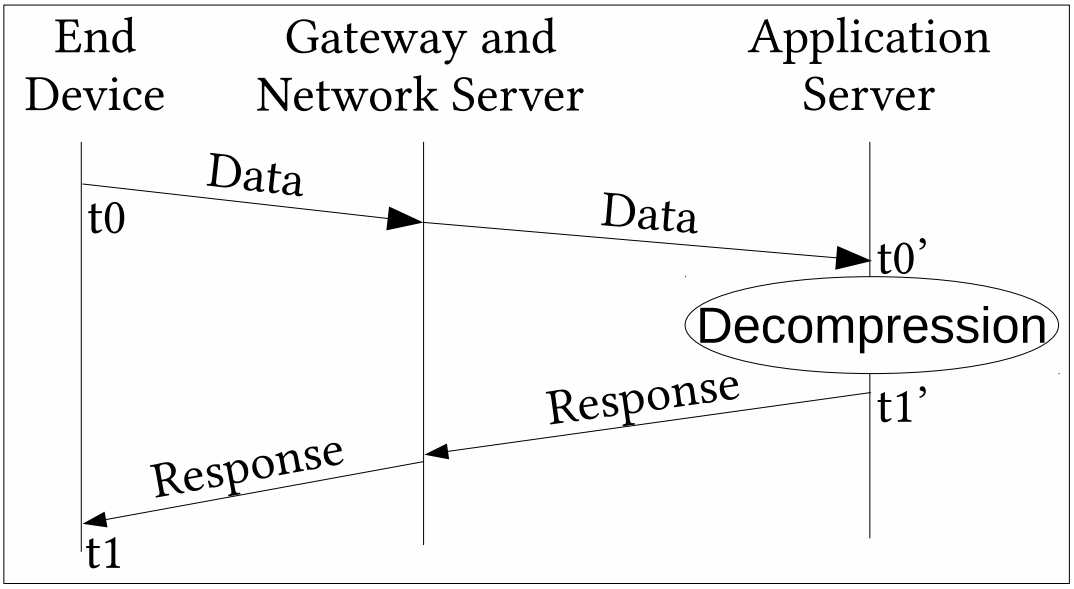


FIGURE 4.5: Message Exchange Diagram (Scenario 2)

- Scenario 2: The second measurement adds the **SCHC mechanism for the communication over LoRaWAN**. The ED sends the data compressed using the SCHC context, and the received data are decompressed using the same context rule stored in a file locally on the AS. We measure $t1 - t0$ (cf. Fig. 4.5) We also measure the AS Response Time $t1' - t0'$. The comparison with results from Scenario 1 allows us to estimate the **decompression time**.

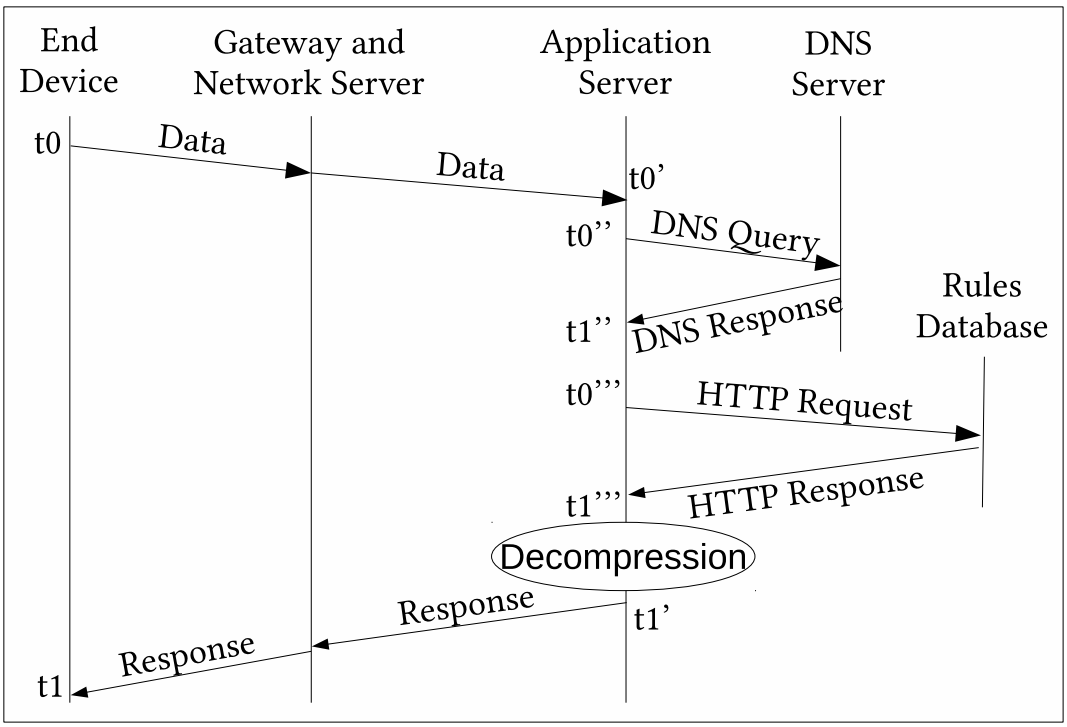


FIGURE 4.6: Message Exchange Diagram (Scenario 3)

- Scenario 3: The third measurement is the key scenario of our study. It aims to

add the mechanism presented at the beginning of 4.2 and illustrated by Fig. 4.3 to provide the AS with the SCHC context that is stored in a remote server. In this measurement, instead of using a locally stored context rule for decompression, the **AS is asked to download the context file from a remote HTTP** server with a request such as "HTTP GET myschcrules.net/DevEUI/RuleID". We measure the total RTT $t1 - t0$, the AS Response Time $t1' - t0'$, the RTT of the DNS query $t1'' - t0''$ and the RTT of the HTTP Request $t1''' - t0'''$ (cf. Fig. 4.6)

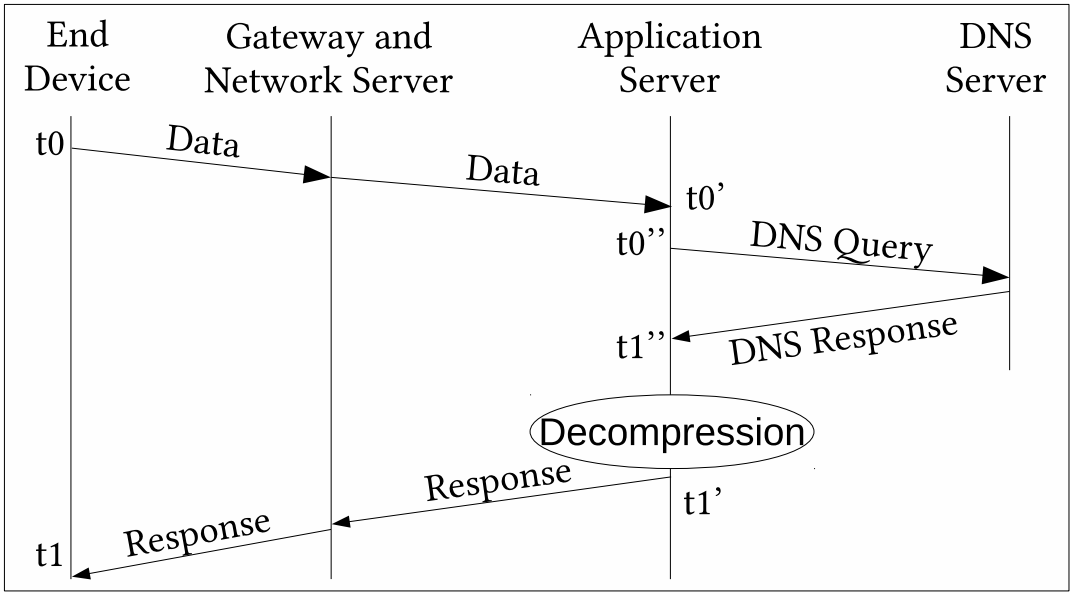


FIGURE 4.7: Message Exchange Diagram (Scenario 4)

- Scenario 4: In most cases, EDs will be static (e.g. water meters) and well known by the AS, so their context rules will always be present in the AS cache of rules. In this case, the DNS is still queried to **check that there has been no change to the rule**, but there is no need to interrogate the HTTP server as the rule is still present in the cache. The third measurement corresponds to this scenario. We measure the total RTT $t1 - t0$, the AS Response Time $t1' - t0'$ and the RTT of the DNS Query $t1'' - t0''$ (cf. Fig. 4.7)

4.2.3 Experiment Testbed

As mentioned earlier, our study is done using LoRaWAN. ChirpStack[318] is an open-source solution to build a ready-to-use LoRaWAN easily. It provides the software components of our infrastructure for the RG, NS and AS.

ChirpStack works with the RG to ensure that the data received from the devices can be relayed to the AS. For our experiment, we chose to connect directly to an MQTT broker and subscribe to the message queue associated with our devices, but MQTT can also be used to monitor the RG or contact all the devices linked to a specific LoRa Application using various topics. ChirpStack AS also offers a REST API, a gRPC API and a web interface to offer multiple ways to operate a LoRaWAN network.

We used **PyCom FiPy** development cards as LoRa-enabled devices, and we made them send SCHC compressed data based on a context over LoRa to a **Multitech Conduit RG** which forwards the data to the **ChirpStack NS**. Then we can retrieve

the data using the **ChirpStack AS** or subscribe to the MQTT broker hosted on the NS to retrieve the data sent and decompress it based on the same context used for decompression. The SCHC implementation used to decompress data is **OpenSCHC** [323]. OpenSCHC is developed by the authors of the SCHC draft as a PoC. It serves as the base reference for other SCHC implementations.

FiPy cards are Class A compliant devices as defined by the LoRa Standard [326]; hence they respect a strict emission/reception schedule. Our experimentation is realized respecting the EU regulations on duty cycle, communicating in the EU 868 MHz frequency, and all communications are done using SF7 considering that, for our experiment, it is the one we expect to include most constraints regarding latency. If our system works without hindering RTT for SF7, it has no reason to hinder the RTT for higher latency SF.

4.3 Experimental results

Fig. 4.8 illustrates the cumulative distribution functions of the AS-side Response Time $t1' - t0'$ with or without SCHC (cf. Fig. 4.4 and 4.5) to show the order of magnitude of the sole decompression mechanism. For this case, we consider that a **locally stored context file is used for the decompression**. The curves show that integrating SCHC adds a few milliseconds to the operations necessary to work on the data independently to the possible delays added by the rule-querying mechanism.

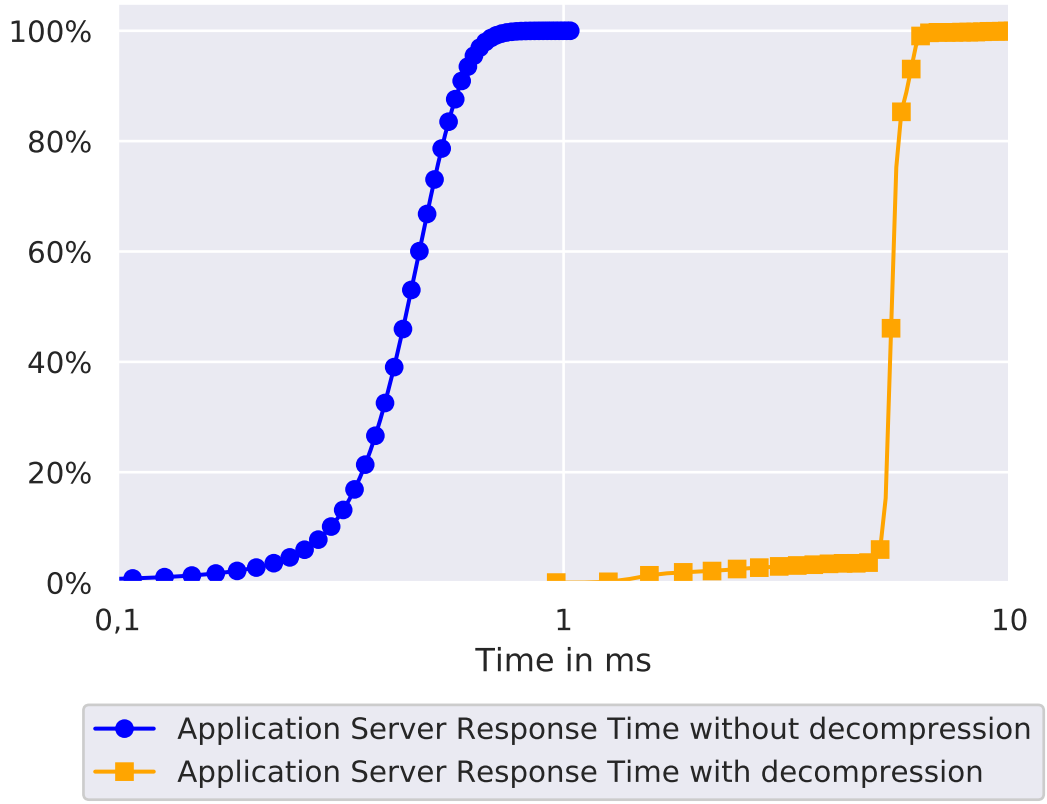


FIGURE 4.8: Cumulative distribution function of the AS Response Time $t1' - t0'$ (in %) against time in ms for Scenarios 1 and 2

Fig. 4.9 shows the cumulative distribution functions of the Server-side Response Time $t1' - t0'$ for all the studied scenarios, thus including also **context remote querying for the non-local solutions** (HTTP, DNS).

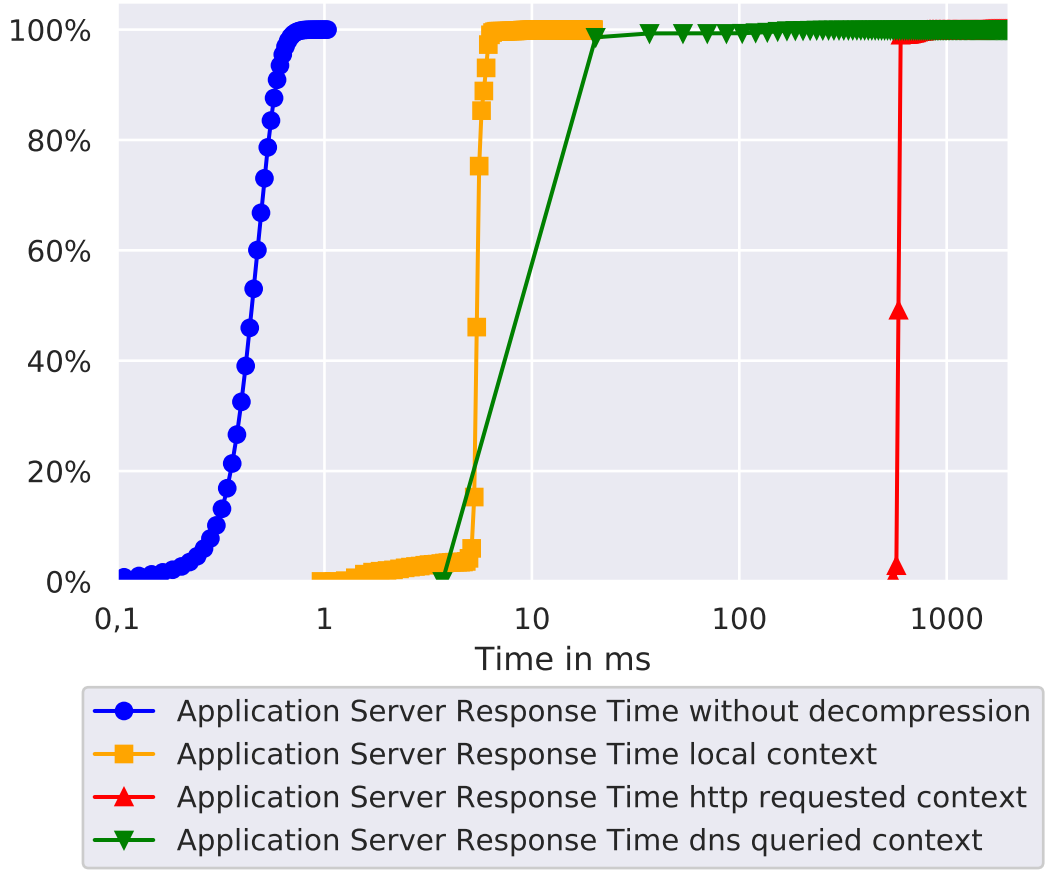


FIGURE 4.9: Cumulative distribution function of the AS Response Time $t1' - t0'$ (in %) against time in ms for all scenarios

We observe that the order of magnitude of the AS Response Time is for the worst-case (HTTP-base mechanism) around 0.6s.

Note that the DNS response time in our case (between 5 ms and 15 ms) is faster than the usual DNS response time due to DNS caching [234] from our local network's DNS resolver. We keep interrogating our resolver with data already in its DNS cache, so the **DNS Response Time is cut down**. This caching will remain in a wide LoRa deployment, but considering the frequency with which the LoRa devices are expected to communicate on the network, the cache will probably be emptied from the necessary data.

In order to provide a more realistic model to study the influence of adding DNS queries in an IoT system, we decided to gather additional data on DNS response time. We used **RIPE Atlas** [327] which is a system that assists in performing Internet measurements through a set of probes available all over the world. RIPE Atlas is a global network of probes deployed under the scope of RIPE NCC. Its 12000 probes enable Internet connectivity testing throughout the globe, resources availability testing, and real-time measurements of the state of the Internet. RIPE Atlas

is a valuable and powerful tool for **troubleshooting**, **monitoring**, **testing**, and **experimenting** over the network. While most of the probes are in Europe, we realized measurements asking for interrogations from all other continents (Oceania, Americas, Asia and Africa) to **test the responses for a single DNS query from multiple locations worldwide**. The measurements performed using RIPE Atlas allow us to determine the DNS Response Time in a more realistic case, as it allows us to query when the DNS cache is expired.

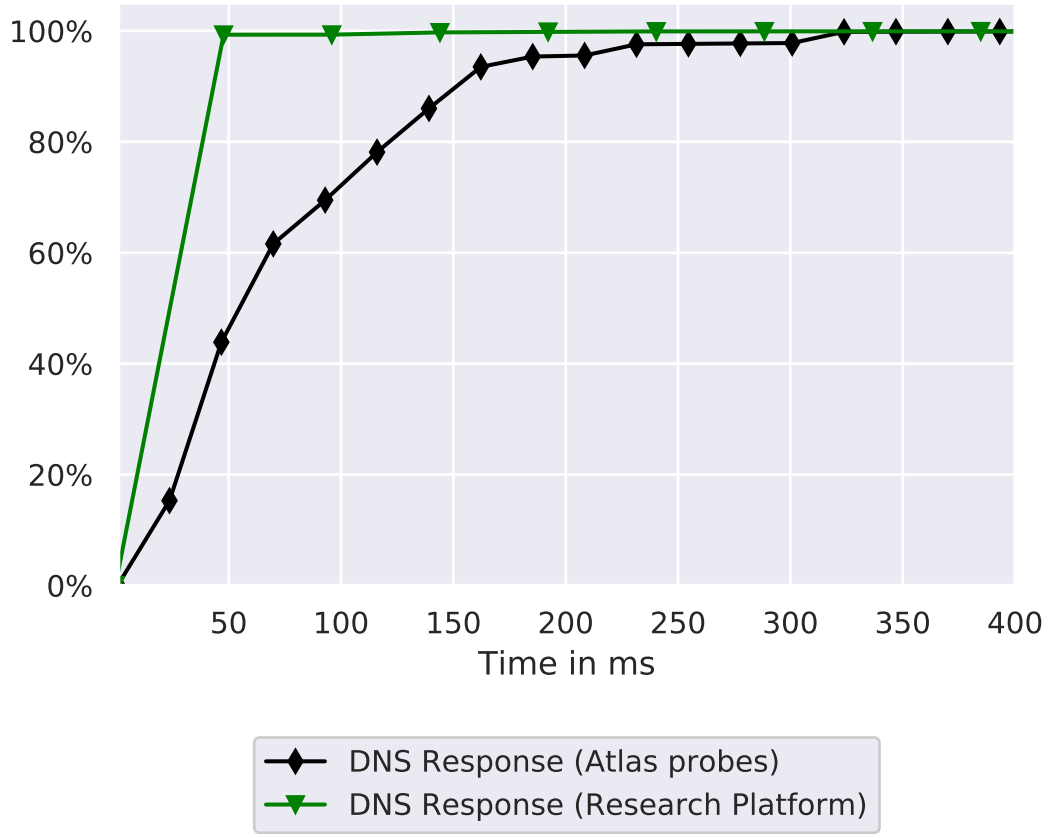


FIGURE 4.10: Cumulative distribution function of the DNS Response Time $t1'' - t0''$ (in %) against time in ms for Scenario 3 compared and from RIPE Atlas [327] Measurements

Fig. 4.10 provides a comparison between DNS response time for our DNS Queries and DNS response time obtained through measurements from RIPE Atlas interrogations. According to this figure, DNS Response will be slower in a real case than with our platform, but a time within 200ms is still **in the margins with regards to the response times we measured for our platform**.

This observation is consistent with data from the literature, [234] provides additional information regarding latency distribution through a study of the Massachusetts Institute of Technology's DNS resolver. In their study, DNS lookups range from a few milliseconds (when accessing from the cache) up to 120 seconds, with most domains resolved within a 1s timeframe (and 70% within 200ms). The study also provides additional information linking latency to the number of referrals, with more referrals linked to more lookups and thus more latency observed.

Our case can be correlated with their 1-referral resolution. In 2001, when the study was realized, 60% of the resolutions were made within a 200 ms timeframe. Considering the evolution of DNS deployments, such as the deployment for massive DNS resolvers such as Google's or Cloudflare's, and the improvements linked to the hosting of DNS zones, such as the massive use of anycast and the development of cloud technologies, a 200ms latency seems coherent.

This timeframe would be further increased with the use of newly developed DoT or DoH, [255] and [256] provide insight by comparing the technologies through the same tool as ours, RIPE Atlas probes. [255] concludes that DoT and DoH indeed increase response time, an observation that can be linked to the use of TCP instead of UDP and the additional cost from encryption. The tradeoff from this loss in response time comes from securing and improving reliability when navigating the Web. Our use case is fairly different from web navigation and does not necessarily require additional encryption, thus relying on traditional DNS seems acceptable. [256] provides additional information regarding DoT resolution with RTTs around 15ms for traditional DNS resolution and RTTs over 100ms for DoT use. Securing DNS resolution with DoT seems to come with a 100ms tradeoff.

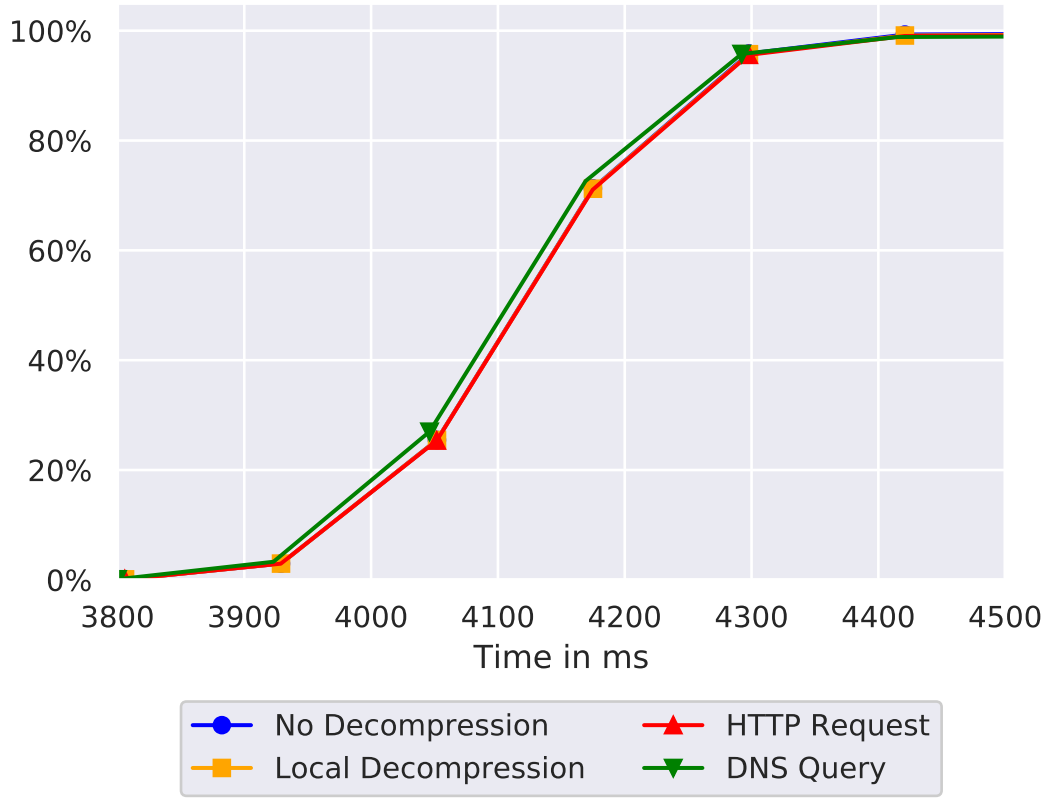


FIGURE 4.11: Cumulative distribution function of the RTT $t1 - t0$ (in %) against time in ms for all scenarios (all the curves are the superposed)

Fig. 4.11 presents the measured U-RTT (From ED to AS, then back to ED) $t1 - t0$. We measure at least about 4s for 99% of the packets transmitted through our platform for all the studied scenarios. Considering the case of LoRa Class A devices [326], a downlink frame from the RG can only be sent during a given time interval called "receive window" (cf. [328] & Fig. 4.12). The RG implementation we are working with

does not allow a frame to be transmitted to the device unless it has been en-queued before the RG receives an uplink frame from the device (cf. Fig. 4.6). The last receive window is opened two seconds after the last uplink frame has been transmitted. It lasts twice the transmission time, which depends on the SF. In our case, we use SF7 and our transmission time is around 100ms. For the majority of our measurements, our total measured RTT is around 4.2s, as illustrated in Fig. 4.13. Note that we use the OpenSource ChirpStack implementation, the reference solution for LoRaWAN OpenSource deployments. We would expect the same behaviour for class B devices, whereas Class C would allow an immediate response and a shorter RTT.

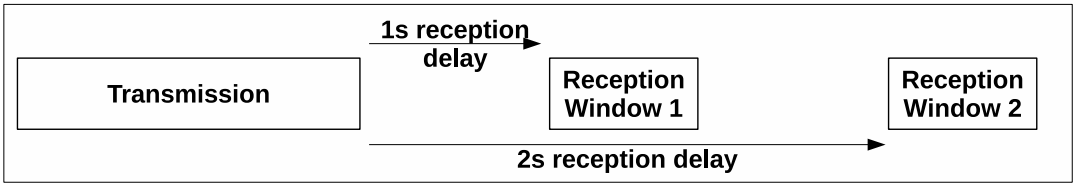


FIGURE 4.12: LoRa Transmission/Reception Windows

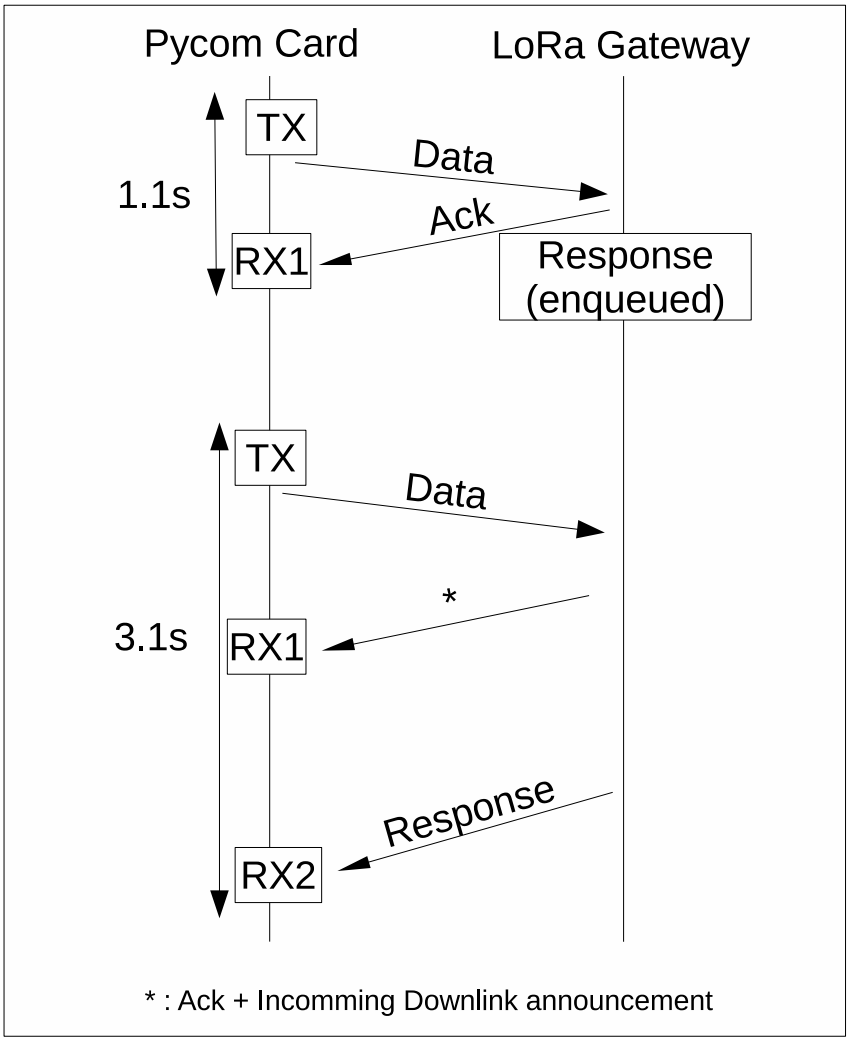


FIGURE 4.13: LoRa Communication Timing

4.4 Conclusion

SCHC can efficiently compress headers from the 44-octets IPv6+CoAP header down to a few octets, which leads to reduce header size up to a 22-factor and enable the use of IPv6 over networks that would be unable to support it, such as SigFox and its 29 octets frame or LoRaWAN which ranges from 59 octets to 250 octets. Using SCHC to send IPv6 packets over LPWAN is proven to be an efficient way to account for the scarcity of radio resources. SCHC is also able to work within a reliable timeframe. Querying time within a large database was not studied and would need additional data regarding actual SCHC usage on LPWAN to provide interesting insight, but when working with a few devices, SCHC can decompress data consistently with a few microseconds; an operation that is almost transparent with regards to other mechanisms specific to LPWAN, such as LoRaWAN's reception delays.

In our experiment, we deployed all the components of a LoRaWAN infrastructure in order to build a SCHC-enabled LoRa network. Because of the expected large number of devices and the variety of possible things profiles, it seems necessary to envisage a mechanism to retrieve SCHC context rules dynamically. DNS is a globally and well-known system that is a fundamental stepping stone when designing a dynamic system.

Thus our method proposes to accompany the SCHC mechanism with a context querying system to support device mobility and network scalability. Using DNS, one can query the context signature within a satisfying timeframe (between 30 and 100 milliseconds) and fall back onto the associated context storage API within 650 ms. In a best-case scenario, the 650 ms delay would be reduced furtherly by caching the information; our Atlas probes measurements lead us to believe that using a DNS-only mechanism and building a context cache would reduce the 650ms delay down to tens of milliseconds. These results concerning the DNS-only system and its performances are consistent with results measured in other studies. Such a mechanism does not hinder the communication as it is kept under the delay of the first reception windows and benefits from the information caching should the answer need a different SCHC rule. Should we need to respond to the device within the first reception window, we are left with around 350ms of data handling in the worst-case to enqueue our response onto the RG.

Regarding LoRaWAN RTT, working with SF7 measurements, we only observe a 4.1s mean RTT when considering the listening window used by the device to receive communication from the RG. This observation is consistent for all scenarios and is easily explained by the prevalence of the delay linked to data reception. We found out that ChirpStack's NS does not handle responding to the device after information treatment unless we consider the Join procedure, which leads us to propose RTT measurements based on sending two information within two successive transmission windows to acknowledge and respond to the data sent from the device.

Problems may arise considering upper-layer protocols such as CoAP. This question was asked at the IETF by C. Gomez and J. Crowcroft in their draft RTO considerations in LPWAN [328] for which the authors signal that "LoRaWAN policies may lead to U-RTT up to 282 seconds in the worst-case" (SF12). SCHC should not hinder CoAP, as packet handling is done within a few microseconds. However, as our mechanism may induce additional treatment up to 650 ms, additional measurements linked to CoAP compression/decompression considering CoAP handling on top of

SCHC remote context querying might be an interesting subject to study in further work. However, considering that "LoRaWAN policies may lead to U-RTT up to 282 seconds in the worst-case", a 650 ms additional delay is negligible by several orders of magnitude.

Actually, according to [329], CoAP message transmission has a default ACK_TIMEOUT parameter which is set to 2 seconds. In this case, the ACK_TIMEOUT has to be adjusted carefully to respect end-to-end delays. In this case, adjusting the delay to handle RTO consideration in LPWAN should prove a sufficient adjustment to handle SCHC remote context querying.

Should one decide to add security to the channel between the NS and the context storage servers, exploiting TLS by using either DoT (DNS over TLS) or HTTPS (HTTP with SSL/TLS) would add another 100 ms to the mechanism, which keeps us within a sufficient timeframe. Unfortunately, it is to note that DNS, or any remote querying mechanism, would be hindered in locations that benefit from a slow or distant connection to the backbone. One such example is Oceania, in which the Atlas probes signal either a significant amount of packet loss or high latency in DNS resolution. Using SCHC without DNS or building a proximity context registry would be the solution in such a rare context. Additional insight regarding storing and sharing data within the DNS will be provided in section 5.5, with a focus on the size of DNS RRs.

Further work regarding the SCHC protocol would require additional data from the actual use of the SCHC protocol. Studying SCHC uses would help define the research direction by contrasting them with the production issues introduced by the protocol. Studying SCHC context rules outside their theoretical construction, but based on actual rules used in production, would help further identify eventual new constraints introduced by the protocol and define useful research direction.

As a matter of fact, improving the compression capabilities of LPWANs is a key concern for the technology. Reducing packet size reduces airtime, which is an efficient way to improve the scalability of LPWAN solutions. Another solution to reduce airtime of LPWANs transmissions is transmission minimization. Our approach with transmission minimization is complementary to compressing header using SCHC. SCHC relies on suppressing and optimizing predictable data within the transmission's header. However, what if the actual relevant data from the payload is predictable. In this case, would it be possible to study transmission minimization paradigms in which compressing communications rely on predicting a device's transmission payload and preventing its transmission if it is unnecessary? That is the subject of the next chapter.

Chapter 5

Network traffic minimization based on Machine Learning predictors

5.1 Introduction

In the previous chapter, we've seen how SCHC improves LPWANs by offering them interconnection with the rest of the Internet, circumventing the high constraints and notably the limited payload size. In LPWANs, traffic compressing is of the utmost importance, our previous approach revolved around header compression mechanisms, this chapter will dive into minimizing traffic by compressing LoRaWAN traffic, more precisely by reducing its data payload, using a ML-based compression scheme.

Many IoT applications consist of monitoring: power grid or water distribution network metering, electric vehicle battery level monitoring, meteorological, temperature, or humidity monitoring. In most cases, the observed time series are highly correlated and can be forecast easily unless unexpected events occur. Thus, it is not necessary to transmit the data in most cases, but only of unexpected events. With a good time-series predictor both on the sensor and on the network backend, the data can be deduced at the backend without sensors transmissions. However, if the sensor, using the same predictor, observes that the measured data is different from the predictor's forecast, if it notes an unexpected event, then the sensor must send the data. With such a mechanism, we can avoid many transmissions and produce highly compressed traffic. Our compression approach differs from usual compression methods, based on pattern frequency analysis, as our studied compression proposal suppresses transmission entirely instead of compressing the payload. However, those two approaches can be complementary, suppressing unnecessary transmission and compressing the remaining transmission payload.

Our goal is precisely to test to what extent such an approach is well suited for IoT, particularly in LoRaWAN. We want to observe the efficiency in network performance (e.g. compression ratio) and power consumption since it is essential to save the batteries of sensors that are expected to have a long life. We also want to test whether our solution is feasible practically by setting up an experiment with actual sensors.

Different predictors may be envisioned, but ML, particularly neural networks, is well suited to model any repeated pattern. Here, we use an LSTM neural network

(as described in [330] and Figure 5.2) for two scenarios: power production and consumption metering and cellular base station load monitoring. We trained the neural network model on a powerful computer, and then we **injected the trained model into the sensors**. We measured the compression ratio and the sensor's electric consumption, considering the transmission and the computation cost. We run our experiments with real measured data and LoRaWAN equipment.

This work also presents insights on LSTMs' accuracy and how the device's compression capabilities are impacted by LSTM accuracy.

In part 5.2, the related works are reviewed. Then, in part 5.3 we present our experiment setup, tools and software, implementation choices and the walls we encountered. Lastly, in part 5.4, we present our results and discuss possible improvements to the system. 5.6 sums up our experiments and conclusions.

5.2 Compression and Machine Learning

The easiest way to reduce data transmission is to delete redundancies or to round them to near values. Sensors often generate time-correlated data. For example, the temperature may vary slowly. Run-Length Encoding (RLE) takes advantage of the adjacent clustering of symbols that occur in succession. It replaces multiple symbols with a tuple that contains the symbol and the number of times it is repeated. The authors of [142] apply Delta Encoding followed by RLE at the end node. In [135] delta compression allows sending only the difference between two consecutive temperature measurements. Data are also usually quantized to round the measures to significant approximations requiring fewer bits for coding. They are often aggregated ([126] or [129]).

Approximating the measurements reduces their size also [127]. One can compress signals by approximating them with auxiliary, more simple functions. Lightweight Temporal Compression (LTC) [150] is an energy-efficient lossy compression algorithm that maintains memory usage and per-sample computational cost in O(1). LTC estimates data points using a piece-wise linear function that guarantees an upper bound on the maximum absolute error between the reconstructed signal and the original one while maintaining a memory usage and per-sample latency in O(1).

Classical compression approaches based on dictionaries or entropy coding have been adapted to IoT, like [145] where a specific dictionary is created for different kinds of data depending on their change frequency.

Transform methods are classical tools for compressing data but may be CPU resource consuming. [120] evaluates several lossy compression algorithms for efficiently storing weather sensor data based on the encoding of temporal changes and three signal transformation algorithms on spatial data. Specifically, they evaluate reconstructed weather sensor data fidelity using Discrete Cosine Transform, Fast Walsh-Hadamard Transform and Discrete Wavelet Transform (and Lossy Delta Encoding). The objective is to provide useful information for minimizing data reconstruction errors, and more importantly, make sure they are within a tolerable range. Chebyshev compression is considered in [121] and [141].

Compressed sensing is a new technique. As stated in [152], in a series of pioneering works by Candes ([153], [154],[155]) and their co-authors, it was shown that when a signal has a sparse representation in a known basis, one can vastly reduce the

number of samples that are required—below the Nyquist rate and still be able to recover the signal (under appropriate conditions) perfectly. This framework suggests compressing the data while sensing it; hence the name compressed sensing. Nevertheless, on the one hand, compressed sensing reduces the number of measurements and the sampling rate. However, on the other hand, it increases the computational complexity of the signal recovery ([156]). The signal is recovered approximately by solving a convex relaxation of a non-convex optimization problem. [134] proposes a unified approach for compression and authentication of smart-meter reading in advanced metering infrastructure. In [133] an algorithm is designed which combines the accuracy of standard lossless compression with the efficiency of a compressive sensing framework. Given the sensor node battery state, the algorithm balances each technique's tradeoff and optimally selects the best compression mode by minimizing reconstruction errors.

Recently, Neural network-based techniques entered the landscape of IoT data compression techniques. In [144], data are compressed by their regression curve obtained from a neural network. In [157], biomedical signals are compressed using autoencoders. These neural networks are three-stage networks with the same input and output dimensions, while the hidden stage has a smaller dimension. Thus, the first stage's output has a reduced dimension compared to the input and constitutes the compressed data.

Another part of transmission compression is header compression, recent work from IETF develops possible ways to improve payload efficiency by compressing packet headers such as ROHC [331], or SCHC [3].

Prediction methods are also used. Neural networks are known as universal function approximators with the capability to learn arbitrarily complex mappings, and in practice, show excellent performance in prediction tasks. In such context, a sufficiently well trained neural network shows better results than more classic approaches[332]. Thus, the authors of [147] train a RNN predictor followed by encoding with a traditional arithmetic coder block using the probabilities generated by the trained neural network. The decompression is performed symmetrically and requires the trained model for arithmetic decoding. In [158] a prediction scheme is implemented on cluster nodes and cluster heads to reduce data transmission. If the measured data corresponds to the predicted one, it has not to be transmitted. LSTMs are proposed to perform predictions. We decided to push the subject further by implementing the algorithm directly on the sensors instead of relying on simulation. Our PoC experiment aims to back these simulations or disprove them should the system prove unreliable.

On the subject of traffic data prediction, some articles propose to use a similar approach using large LSTMs such as [333]. Their multi-feature approach allows them to correlate data and obtain interesting results regarding traffic prediction. We hope to obtain similar results with our curves as we work with network traffic. However, our approach differs in our decision to focus on the effect of predictions on transmission: improvements to LSTM capabilities are out of our scope.

This approach was also tested with energy production forecasting. LSTMs are presented as a possible candidate for energy production forecasting in [334]; the solution seems adaptative enough for our approach to be reliable enough when using **LSTM as a forecasting tool**, [335] validates this approach by comparing it to other neural networks.

Thus, many compression techniques appeared for many years. Nevertheless, if the "classical" methods present efficient compression ratios, they do not avoid transmitting data. Actually, periodically a sensor senses data, may compress it and then send the compressed payload. Nevertheless, compressed data payloads (plus header) are still sent. New neural network-based techniques appeared, and they avoid sending data at all in some situations where the prediction is good, but to our knowledge, they have not been tested with actual data and on real equipment. The goal of our test is precisely to propose a validation in real conditions.

For this approach, the prediction algorithms used rely on the approach from[335]. We aim to have a reliable data prediction based on an LTSM neural network and run the predictor on both the sensor and the network infrastructure. As we can expect prediction performance reliable within a 10% **Mean Absolute Percentage Error** (MAPE) (equation 5.1), we might expect a complementary compression coefficient up to 90% for our experiment. Such a compression ratio would allow us to build sensor networks where each device consumes less bandwidth, thus further improving the scalability of LPWANs solutions.

MAPE in this experiment was calculated as follow:

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{C_i - M_i}{C_i} \right| \tag{5.1}$$

With M_i as Measured value, C_i as consolidated value and n as sample size

With our experiment, we test various **neural network sizes** and **transmission thresholds** to measure how these parameters might influence the **compression ratio** of our device's transmissions.

5.3 Experiment

We embarked an LSTM algorithm similar to the ones studied in [158] and [335] onto an electronic device as a way to test the experimental feasibility of these solutions. These articles propose to use LSTMs that are sufficiently simple to be implemented and embarked onto devices.

The devices used for this experiment are STM32L476 [336] as measurement and calculation device which generates prediction data and compares it to measurements, Semtech's SX1276MB1MAS [337] as LoRa transmission board, and STM32 Nucleo Expansion Board [338] to measure the energy consumption of LoRaWAN transmissions.

The datasets we used are occupancy data of cellular base stations for the 1st dataset and power consumption of a smart building for the 2nd one, in function of the time. We used the 1st dataset for all experiments except for those presented Figure 5.5.

For these experiments, all experiment are realised with 16-bits operations except for 5.7 where 32-bit operations are realised through simulation in Python using the TensorFlow library and 8-bit operations are realised onto the device. All experiments with fixed threshold are run on sensors with transmission minimisation and all variable thresold are simulated.

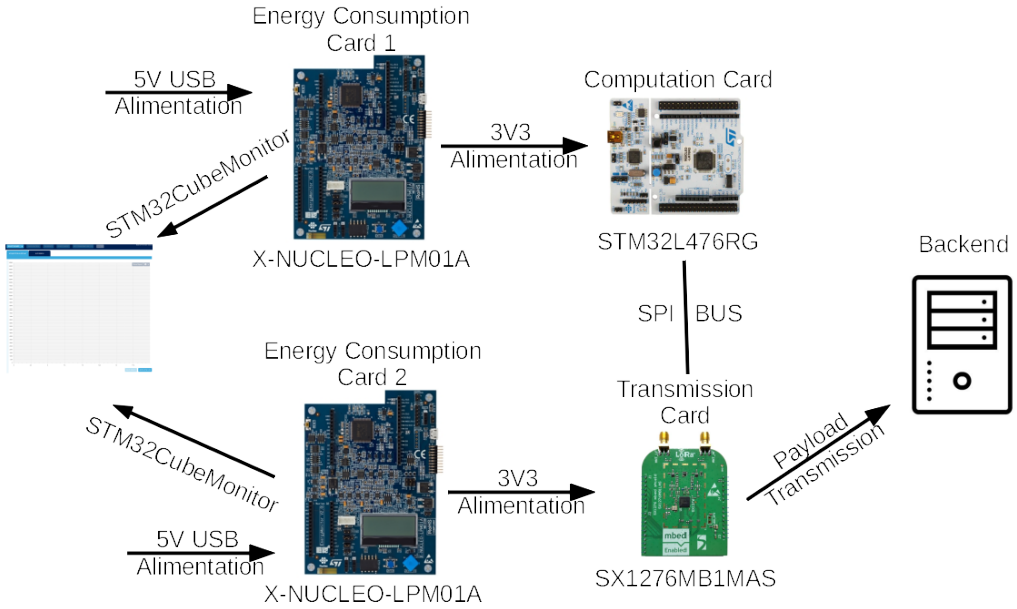


FIGURE 5.1: Experimental testbed

5.3.1 Hand coding the neural network

Initially, we thought of basing our experiment on EdgeImpulse [339]. EdgeImpulse is an easy-to-use and well-documented framework to generate ML models from actual sensor data and automatically embed them onto the device. EdgeImpulse was a strong candidate to support our experiments on sensors. Unfortunately, the framework offered from EdgeImpulse did not support LSTM functionalities, thus was not adapted to our use case.

LSTM are a special kind of RNN capable of handling short-term memory while keeping tabs on long-term information. LSTM consists of a repeating module that consists of five gates detailed on 5.2, and keeps part of its long term memory with its cell value (c_t).

The first sigmoïd gate (a sigmoïd is a $\sigma(x) = \frac{1}{1+e^{-x}}$ function) serves to amend the cell state by forgetting a part of it based on the inputs and the weights. The associated operation is as follow:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.2}$$

where W_f and b_f are the weights for the forget gate, determined during training, and h is the cell's previous output (which can be a vector considering multiple hidden cells in parallel).

Then we input new information into the memory by combining the sigmoïd input gate with a cell candidate determination function input gate is

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.3}$$

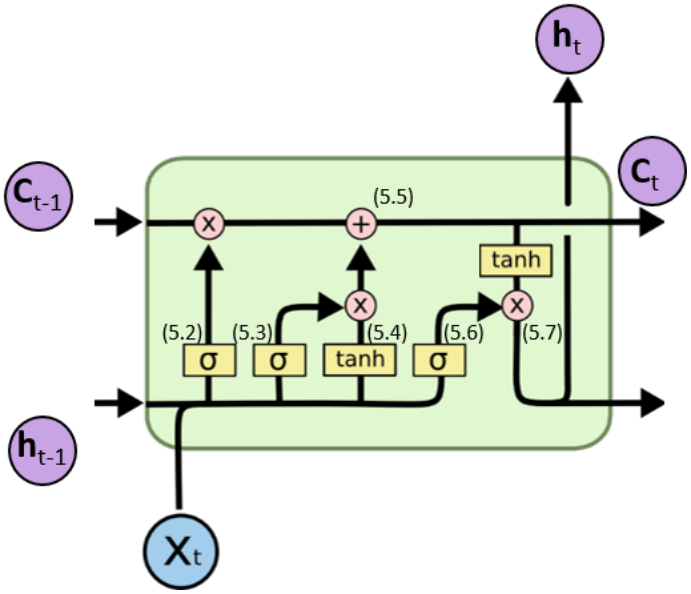


FIGURE 5.2: LSTM Network extracted and reworked from [340]

and cell candidate is

$$\tilde{c_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5.4}$$

These operations flush part of the input information and select which part of the data is to be kept as long term information by adding it to the amended previous cell value.

The new cell state c_t is determined by combining the previous cell state c_{t-1}, forget gate output f_t, input gate output i_t and cell candidate $\tilde{c_t}$, and will pass onto the following iteration of the LSTM. The associated operation based on the previous variables is as follow:

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c_t} \tag{5.5}$$

Finally, the cell outputs its results by combining an output gate and the new cell state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5.6}$$

$$h_t = o_t \times tanh(c_t) \tag{5.7}$$

h_t serves as output to the system, and input to the next iteration for our calculation.

As mentioned above, EdgeImpulse did not support this kind of LSTM. By studying the source code of the program generated by EdgeImpulse, we constated that the basic library used to execute the ML algorithm was the TensorFlow Lite (TFLite) [341] library. So, we built an mbed OS firmware that embarks the **TFLite exported neural network** and transmits data based on the predictions but could not exploit

the strength of the TFLite for microcontrollers library as the LSTM operations are not supported yet.[1]

We trained our LSTM network with TensorFlow [342] and exported the LSTM weights and parameters necessary to our implementations (weights, bias and hidden layers' state). We ported the LSTM code successfully onto basic STM32 boards. We **injected the parameters** into our **own implementation of the LSTM** network, developed in C, and **embarked the LSTM** network directly on the sensor.

Our implementation is available following [343]. This implementation was built thanks to the explanations from Christopher Olah [340], and the following tutorial on weights and parameters extraction [344].

5.3.2 Dual prediction with LSTM

The usual approach with LSTM is to use measured values as input to the system and obtain a predicted value based on these measurements, as proposed in [345]. A modified LSTM architecture is proposed hereafter as this approach does not fit with dual prediction since the backend (i.e. the network side receiving the traffic) does not have access to the sensor's measurements. Our experiment relies on a **dual prediction of data** to reduce transmission. It relies on two types of data: On the one hand, we have measured values from sensors, and on the other, we have calculated (i.e. predicted) values. Our LSTM needs to keep calculating based on values on both the backend and the sensor as long as transmissions are unnecessary. Actually, as long as no data is transmitted, the **backend has only the calculated values**, as the backend realizes the same ML-based calculations as the sensor, not the measured ones. Thus, the **backend must run the LSTM** by re-injecting these calculated data into the LSTM, and, consequently, the device must do the same to check to which extent the data predicted by the backend is far or near the data just measured. When transmissions occur, we can recalibrate the LSTM and use the new transmitted value as a new baseline for calculations.

The firmware we developed is then flashed onto an STM32L476 card to exploit the capability of LSTM combined with LoRaWAN transmissions. This algorithm, built on **mbed OS**, uses an LSTM Neural Network to predict a theoretical value at a given time. It compares these theoretical values to experimental measurements at the corresponding time. We define a threshold that determines a transmission policy: if the experimental measurements differ from the predicted value within a given margin, the transmission is not realized. However, if the experimental measurements are too far from the predictions, the data is sent over the air from our LoRaWAN device to our LoRaWAN backend. This **threshold** might be **fixed** as in Figure 5.6, or we might study the effect of **changing the value of the threshold** through simulations such as with Figure 5.4 which studies the **compression ratio** one can expect with this system when picking various threshold values. Such a data transmission policy may allow us to reduce the band usage for our device.

We plug our transmission card (SX1276MB1MAS) into an STM 32 Nucleo Expansion Board to monitor its energy consumption using STM32CubeMonitor [346]. We do the same with our STM32L476 card in order to study the overcost of running the LSTM.

[1] We discussed this issue with contributors from both EdgeImpulse and TFLite for microcontrollers and will do our best to carry on with this work and use it to contribute to the support of LSTM capabilities on TFLite.

TABLE 5.1: Comparison of the mean energy consumption of the calculation card and its variance, with and without LSTM-based compression (in Watts)

With Machine Learning		Without Machine Learning	
Mean value (W)	Variance	Mean value (W)	Variance
$6.31 * 10^{-4}$	$7.57 * 10^{-5}$	$7.76 * 10^{-4}$	$7.61 * 10^{-5}$

On the backend side, we monitor data reception and aggregate the data predicted from the neural network on the infrastructure side with the data received from the sensors, allowing us to plot on a graph the **combination** of the actual **measured data**, for which we consider that the information is 100% reliable, and the **predicted data** which was inferred by the neural network and not disproved by sensor transmission (which is reliable up to a certain threshold). The effect of this threshold will also be studied. In this experiment, our LoRaWAN RG is accessible through SF7 communications and is a part of TheThingsNetwork [347] community LoRaWAN Network. Figure 5.1 sums up our experimental setup and illustrates the various hardware components we use.

Alongside this real experimental setup, we studied the effect of changing the system's variables through extensive simulations allowing us to shorten the experimental exploration, find interesting parameters for our embedded experiment and confront simulation results to experiments.

We also studied the **system's reliability**. We defined the system's reliability as the **MAPE** of the data perceived by the backend compared to the real values. This reliability study investigates the consequences of a **bounded lossy compression** on the values obtained at the system's output. Our compression is lossy because the data recorded by the backend is not the measured one but the predicted one as long as the predicted one is within the tolerated threshold interval, and thus defining a **threshold** means we study the **tradeoff** between accepting a given **error** on our data and improving the **compression ratio**. Our compression is bounded because we set it back to the actual data if the loss exceeds the given threshold.

With this experiment, we aim to evaluate:

- the energy cost added by the ML-based compression scheme at the device side;
- the energy saved on the transmission card thanks to data prediction;
- the compression ratio expected with regards to a given neural network size and data prediction threshold;
- the bounded loss introduced by the solution compared to transmitting all values;
- the impact of quantizing the weights of the neural network predictor by running these experiments with quantized parameters instead of floating-point numbers to improve neural network complexity, memory size and finally energy efficiency.

TABLE 5.2: Comparison of the mean energy consumption of the transmission card and its variance, with and without LSTM-based compression (in Watts)

With Machine Learning		Without Machine Learning	
Mean value (W)	Variance	Mean value (W)	Variance
$5.48 * 10^{-4}$	$4.10 * 10^{-5}$	$9.87 * 10^{-4}$	$7.12 * 10^{-5}$

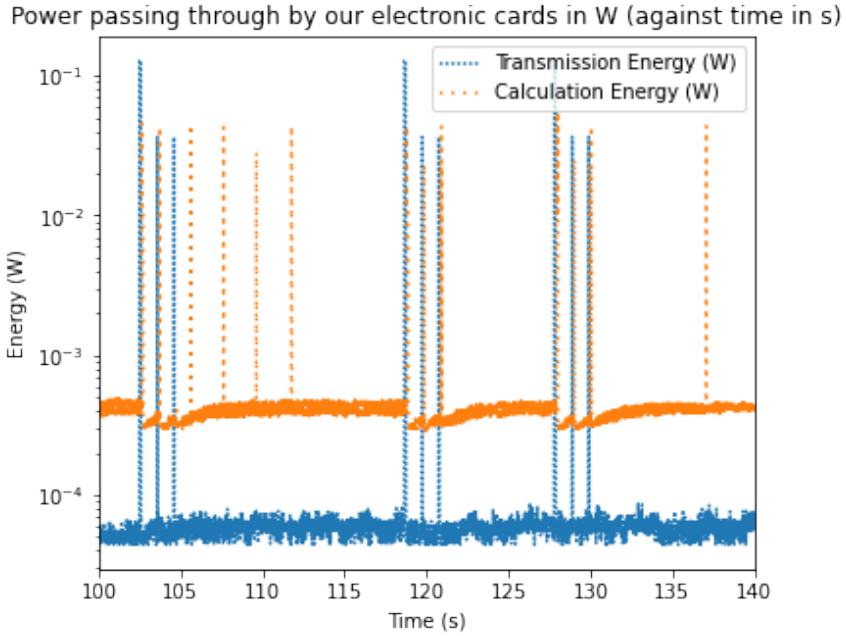


FIGURE 5.3: Energy (in W) passing through the calculation card and the transmission card (Sample)

5.4 Discussion

5.4.1 Energy

The comparison of the consumption of our cards (Table 5.1 & 5.2) shows a save of around 40% on transmission cards. Considering IoT systems similar to our own with measurements stating that calculation and transmission consume about the same, this would represent **savings** of around **20%** on both **transmission** and **battery life**. Considering the card we are using, an LR6 battery with a 1200mAh charge would power our device for around ten months without embedded ML calculations and about a year with ML calculations.

Figure 5.3 presents the energy passing through both our network card and calculation card, measured using **STM32CubeMonitor**, which permits us to measure and log instantaneous consumption for our device, in function of the time. The device lifecycle follows a two-step routine. Most of the device's life is spent in a **sleeping** state with low energy consumption. Here in our illustration, the device's sleeping state is around 9s long to respect LoRaWAN's duty cycle. The device will, **exceptionally** or **regularly**, **transmit data** based on its measurements. Transmitting is the other step in the routine. Transmissions translate in power consumption as **three transmission spikes** corresponding to data emission and the opening of two LoRaWAN

listening windows. A residual energy consumption of about 4.10^{-4}W can be noticed for the calculation card, while it is about 5.10^{-5}W for the transmission card. The calculation card embarks a dedicated OS which requires more permanent consumption. Irregular energy spikes can be observed for the calculation card, which is due to OS eventing. Our ML algorithm directly results in transmission spikes. Except for the operations realized by the OS on the calculation card, no transmission means no power spike, which leads to less power consumption as a whole, a result that can be observed on the transmission card's power consumption. We would observe regular power spikes with regular transmissions, but with our method, these **spikes are completely cut off**.

5.4.2 Compression and Mean Absolute Percentage Error

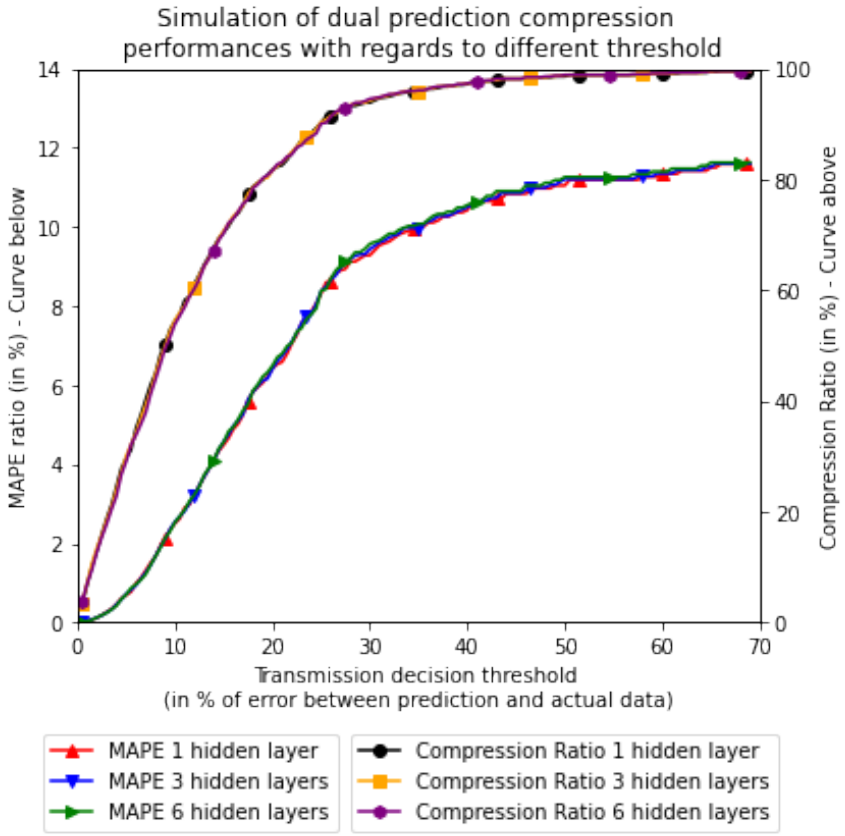


FIGURE 5.4: Compression ratio and mean absolute percentage error with regards to neural network size and precision threshold

Figure 5.4 presents the **MAPE** and the **Compression Ratio** we can expect with regards to the size of the neural network and the decision threshold. We observe that, as expected, the compression ratio improves with the threshold but that the MAPE worsens. The consequences of the **number of hidden layers** is **not significant**.

Experimenting with different datasets (Figure 5.5) show how the performances of the neural network in its prediction greatly influence the quality of the compression scheme. With a better overall MAPE, one might achieve around 60% compression accepting as little as 1% error in its transmissions.

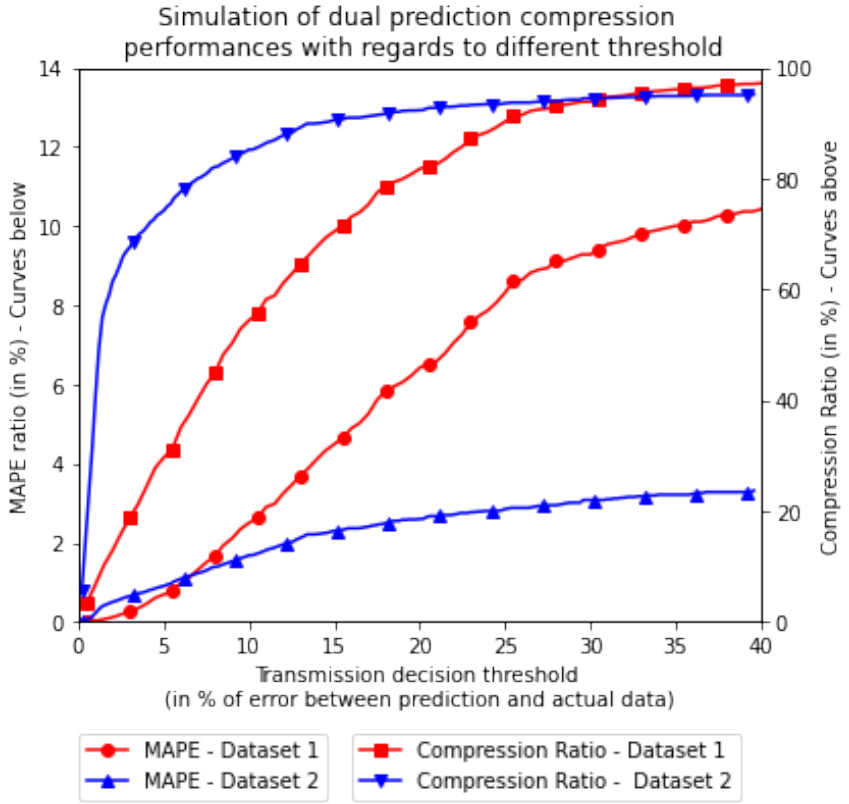


FIGURE 5.5: Compression ratio and mean absolute percentage error with regards to precision threshold for different datasets

The questions that come with these curves need to be addressed directly by the user. A user with concern with precision will prefer a lower MAPE, thus obtaining a lower compression ratio. If a user accepts a 1% error on its global data, setting its transmission threshold around 8%, He would end up with a 30% compression ratio for our first dataset and 85% compression ratio for the second one. Accepting more errors would permit compression ratios up to 90%. We note that the compression ratio is low with a strict threshold, but remember that contrary to classical compression methods where at least a header is sent, no packet is sent with our method when we compress. Thus, for a strict threshold of 10%, we decrease the overall traffic by one packet over five (20%) while keeping a global error on our overall data around 2%.

5.4.3 Backend considerations

Figure 5.6 presents a comparison between the measured time-series as transmitted without ML and the calculated time-series improved with transmissions. The **Calculated Data** curve consists solely of data calculated by the LSTM neural network. The **Measurements Data** curve corresponds to our LSTM target data, and its value end up transmitted should the calculated data end up being too far from the measurements. Finally, the **Combined Data** is the data curve as seen on our backend-side: the calculated data improved by the measured transmission should the two curves differ above a given threshold.

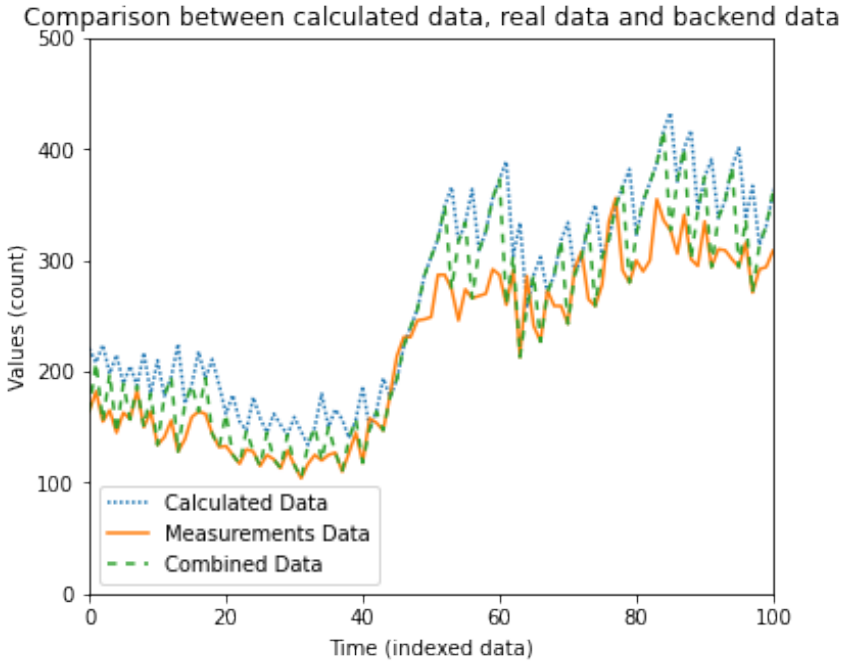


FIGURE 5.6: Comparison sample between calculated data, reference data and data perceived by the backend

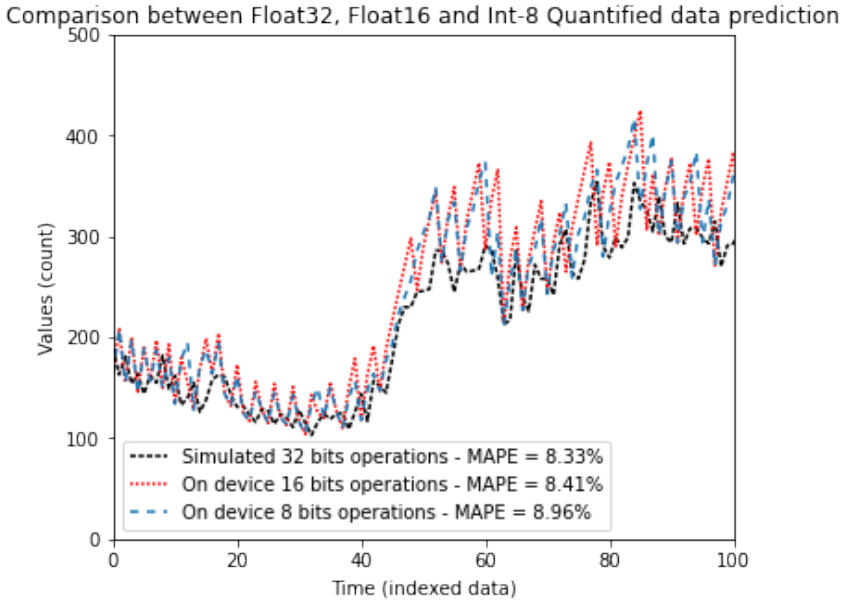


FIGURE 5.7: Float32, Float16 and Int-8 Quantized LSTM forecasting

5.4.4 Quantization

Figure 5.7 presents our backend-side time-series as a function of the time index, combining calculated data and received measurements. The three curves on this figure differ by the number of bits necessary to code the LSTM weights, hidden layer parameters, cell state and input data. The dotted blue curve is obtained by running the dual prediction algorithm in a Python simulated environment and operating with 32-bits-encoded floats. The plain orange curve is obtained by directly running the dual prediction algorithm on the device with an LSTM operating with

16-bits-encoded floats. The dash-dot green curve is obtained by directly running the dual prediction algorithm on the device with an LSTM operating with quantized parameters encoded on 8-bits integers.

Measurements of the compression ratio for the three above curves show no significant degradation between the three systems (the compression ratio almost does not change and remains around 70% for the three curves). 8-bit quantization is a well-documented solution to reduce the operations' complexity while working with neural networks on constrained devices. This also proves to be **confirmed** with **our implementation** of LSTM. Efficient quantization is essential when working with Neural Networks; it **reduces** the **complexity** of the operations, which might, in turn, allow for **savings in processing power and battery life**. Further works would be necessary on quantization efficiency once the LSTM operation is ported to the TFLite for microcontrollers library.

5.5 Storing and sharing ML weights

When working with dual prediction, synchronizing data between backend and device is crucial should one want to exchange information regarding its own neural network. In a similar way that we did in 4.2, we aim to exchange Machine Learning information on the backend side using the DNS infrastructure. Such a solution would allow to benefit from the DNS's deployment, security and trust network to address issues such as keeping the information available, easy-to-use and exchanged using standard protocol.

This section provides insight on possible solution for storing and exchanging the ML weights using the DNS, its infrastructure and extensions, based on different use cases depending on the size of the neural network, but also on possible different IoT network topologies.

5.5.1 Classic DNS use

A first possible scenario for storing and sharing weights would be to host the weights directly on the DNS infrastructure. This first scenario is the same as the Context rule DNS storage from the previous chapter. Thus the only question for storing rules within the DNS would be the size of the neural network.

A DNS TXT RR consists of records with a total size of up to 65535 bytes. However, as mentioned in [5]:

> "the total size of a typical DNS-SD TXT record is intended to be small – 200 bytes or less. In cases where more data is justified (e.g., LPR printing [BJP]), keeping the total size under 400 bytes should allow it to fit in a single 512-byte DNS message"

thus a DNS TXT RR with this paradigm ought to be of reasonable size.

Each ML weight can be encoded on 24 bits (3 digits, 8 bits per digit) thanks to 8-bit quantification, as a DNS TXT RR stores bytes of text, thus would store the ML values as text instead of using more optimized space, but conversion to another base could be considered to reduce storage size.

An LSTM network number of weights can be calculated based on its number of cells.

The calculation for a simple, single-input, network is as follow:

Let H be the number of hidden cells within the LSTM.

Number of weights = $4 * H^2 + 11 * H + 1$

The details for such calculation is as follow:

- $4 * H$ values for the weights associated with the input value
- $4 * H * H$ values for the weights associated with the hidden layers
- $4 * H$ values for the LSTM bias
- $2 * H$ values for the initial state of the hidden layers and the initial state of the cell
- H values for the dense layer's weights
- 1 value for the dense network's bias

Using this 24 bit "standard" for ML values, a simple LSTM network with 1 hidden cell within the LSTM should fit within 48 bytes, thus would fit within the 200-bytes classic DNS message.

The 200 bits limit is attained by using 3 hidden cells within the LSTM (210 bits), and the 400 bits is reached with 4 (327 bits).

Thus heavier neural networks would hardly fit within a 512-bytes message, thus needing additional messages for which the DNS would hardly fit. Using a different encoding for the values, changing the base up to base 36, for example, would allow for LSTMs of size up to 6 hidden layers (7 would be within reach as it fits within 548 bytes).

For even heavier networks, the DNS would prove non-competitive and requesting data from an API would prove better and allow for more functionalities.

5.5.2 Using DNS and APIs for heavier networks

We can design a fallback mechanism for the backend for heavier networks based on a common API for ML models.

Designing such an API model would allow additional functionality to the user, such as generating the neural network via a web interface based on its own time-series.

This solution would help popularise ML transmission minimization by providing to its user a pre-coded compression source code, directly embeddable onto its sensors, and would take care of provisioning both the ML parameters' API and provision the DNS to reference the API location.

Additionally, the time-series provided by the user could help to gather data to improve the neural network training, thus exploiting each individual time-series to improve the overall system performances and provide better solutions to its user over time.

5.5.3 Exploiting DNS-SD paradigms in mesh communications

Considering IoT networks, designing the system to allow communication directly between devices would help improve the overall performance.

The LoRa modulation allows for such a communication solution as it can work with mesh topologies. Using ML compressed communication would allow to upscale LoRa mesh networks by limiting interferences between communicating devices.

However, how would a device transmit the details for its ML model to its peers?

We propose to learn the lessons from the DNS-SD paradigm in a dual connectivity infrastructure to support the discovery and advertising of ML model within a given local coverage.

In this case, a LoRa device would transmit its ML data to its peers as a service advertisement which would be saved by its peers to process its time series and follow the variations from its parameters, increasing the knowledge of its peer within the network.

In the case of a newly arrived device, the overall ML parameters could be forwarded from the RG, which could serve as a DNS proxy as described in [273].

The ML parameters would need to be encoded as base-64 values, allowing for easier transmission of the 8-bit quantified values instead of fitting to the 3-byte-per-value, text-readable constraint from the DNS TXT record.

Additional information can also be advertized through DNS-SD, such as the nature of the sensor, its version, its functionality, or listening cycle, making DNS-SD an interesting candidate for weight advertising in mesh communications.

5.6 Conclusion

We built an experimental testbed to check the capabilities of on-boarding LSTM algorithm on-sensor to forecast data, achieve dual prediction, and eventually compress data traffic and save energy. An LSTM algorithm was developed and integrated into small, constrained hardware to obtain these results; its source code is accessible following [343]. Our findings show that it can efficiently minimize traffic while preventing non-relevant transmissions to occur with a significant impact on energy consumption. We observed the impact of the neural network size and the decision threshold on the compression ratio and the MAPE. Our system allows efficient compression while keeping the user within a reasonable error margin. It can be customized depending on precision and compression tradeoff requirements. We also check the impact of the quantization of the LSTM parameters because of device constraints and decrease the algorithm's complexity. We observed no significant degradation in the system when using 8-bit quantization.

Compressing data with this kind of Bounded Lossy Compression allows to expend battery lifetime depending on the accepted margin of error. Our results show an excellent compression ratio compared to the state-of-the-art. Note that our scheme avoids sending any data while classical compression mechanisms at least send a frame header every time some compressed data is sent. Moreover, an extensive energy consumption study proves that our algorithm saves an important energy ratio that can be used in further communication.

Storing the ML weights in the DNS seems feasible once we run the number seem possible. We presented three approach for storing and sharing the ML weights

within a network with a straightforward approach based on the DNS, an accompanying approach combining APIsand DNS and an autonomous approach based on DNS-SD paradygms.

Further approach would consist of selecting multiple features on multiple values to attain more precision in the calculation with a more complex recalibration. Works are carried by contributing to the actual TFLite community to propose a complete port of the LSTM libraries from the global TensorFlow project to the TFLite for microcontrollers community.

Chapter 6

Conclusion

This thesis presented various approaches to improve interoperability and performances of LoRaWAN-based IoT solutions by leveraging the existing DNS infrastructure as a common ground for application developers and researchers. The DNS is a known protocol for the Internet community, with various open-source implementations for clients, servers or within frameworks. Tools were deployed to measure its performances and modulate its use. Its community is open and proposes new use-cases and improvements that keep the DNS community up to date with new developments, protocols and network paradigms developed by the industrial world or the research community.

Within the IoT ecosystem, LoRaWAN is flexible; backed by an ever-increasing community of industrial stakeholders and a newly-created official academic community, the LoRa Alliance and its LoRaWAN protocol poses as a major actor of the IoT ecosystem. The evolutions from the discussions of the Alliance are closely followed by LoRaWAN applications developers as the reference LoRaWAN open-source Stack, ChirpStack, encountered 17 minor releases and 1 major release within the last 3 years.

Current IoT applications encounter, with their latest development, the same issues as the Internet. IoT technologies are riddled with scalability, interoperability, mobility and roaming, transmission efficiency, availability, reliability and other security issues such as trust and privacy. The DNS contributes to solving many of these issues on the Internet, hence our interrogation on possible improvements to IoT systems backed by the DNS infrastructure.

This thesis studied IoT systems regarding the following key aspects: Naming, Roaming, Header Compression and Payload Compression. This study did not aim to embark DNS protocols onto sensors but instead to use DNS on the infrastructure side to support IoT improvements. This thesis presented experimental work on LoRaWAN regarding various scenarios to test IoT solutions, applications and use-cases. This experimental approach introduced additional constraints such as working with reference implementations of the solutions, generating actual IoT traffic for measurements and analysis, respecting airtime constraints or device lifecycle.

Experimenting with roaming requires an interconnection agreement between network operators, usually based on a 'One-to-One' interconnection or by building an interconnection 'Hub'. We exploited a federated approach to the IoT interconnection by proposing the IoTRoam architecture, federating different organizations to allow flexible mutual authentication and authorization between any backend element in

roaming situations, without a direct and explicit roaming agreement (by interconnecting Network, Application and Join Servers between operators). The interconnection agreement is implicitly given when the organization joins the IoTRoam federation.

By experimenting with LoRaWAN, our architecture proposes a solution that considers the constrained characteristics of IoT environments. Our approach to building our roaming architecture was to use the combination of the DNS infrastructure and a PKI to build a secure open roaming infrastructure accessible to public and private LoRaWAN operators. We leverage the possibility to freely set up private LoRaWAN networks. We designed, built and deployed a proof of concept architecture to test the Roaming capabilities offered by the ChirpStack solution and test roaming between private and public LoRaWAN. The infrastructure was validated by testing LoRaWAN connectivity for devices in a roaming context by studying various onboarding scenarios, measuring onboarding time and communication delays.

We studied the consequences of caching and prefetching DNS information with mobile devices in a city through simulations of communications between mobile device and IoT infrastructure. DNS prefetching is an efficient tool to reduce on-the-fly DNS queries necessary for devices communication. Prefetching the information on nearby antennas can completely prevent DNS queries by performing them in advance around the closest antennas, but at a cost, as devices request more antennas, especially in a highly mobile environment.

Our combination of an ML predictor and prefetching allow for an interesting reduction of DNS requests realized compared to a standard caching-only solution and a reduction in the number of gateways realizing prefetching operation compared to soliciting the closest nearby antennas. Using DNS allow us to exploit its strength as a known distributed database solution to fill localized caches to provide information as soon as needed as well as purge it through time when mobile devices leave the antennas' coverage.

Our simulations were realized within the frame of providing roaming connectivity information to devices, but could be applicable when querying other information necessary for device communication such as certificates stored with DANE, compression parameters or any device specific information stored in the DNS such as pointers to additional servers.

We built a viable and operationally feasible infrastructure that could be integrated into existing IoT infrastructures with minimum changes. We followed the WBA guidelines for open roaming and satisfied the requirements outlined by employing open standards used on the Internet to achieve our vision. We deployed a PoC infrastructure and provided all the necessary building blocks so that the community could make use of them. These experiments lead to three adopted Change Requests after submission to the LoRa Alliance.

We discussed the IoTRoam initiative with several institutions to run interoperable testing using the federated platform; running additional tests with these institutions would help us study the impact of heterogeneous backend infrastructures and their effect on the quality of the communication channel and would also allow us to gather additional data on the impact of DNS complete resolution on the LoRaWAN/IoT traffic. As the objective is to interconnect networks using different IoT technologies, the next steps consist of testing roaming interoperability with NB-IoT, 5G or Wi-Fi. For ED onboarding, we are also working on integrating DANE with DNSSEC since

the certificate data itself can be stored in the DNS, possibly replacing or completing the PKI.

Complementary to our work with interconnecting IoT infrastructure, we experimented with SCHC remote rules management and sharing as a solution to improve the scalability of LPWANs solutions. Providing a way to exchange rules information between backend would offer new possibilities to provide more flexibility to the network, thus improving the flexibility of IoT solutions in contexts such as roaming devices. The proposed mechanism exploits the DNS infrastructure as a solution to leverage DNS performances and improve rules querying capabilities within the network. Unfortunately, rules are too heavy to be embedded directly as DNS Resource Records; thus, a fallback mechanism was designed based on APIs and exploit DNS caching mechanism to store rules identifiers and version numbers. DNS, as an optimized, hierarchical and distributed database, could assist in identifying the location of the server where the context rules are stored feasibly on the Internet. Hopefully, using such a mechanism would allow for a seamless transition, from pre-configuring the information needed on the backend to building it dynamically, on the fly, based on actual needs when operating the network.

DNS would prove an efficient solution to introduce more flexibility and improve scalability when using SCHC. Our solution would provide open access to SCHC parameters as a way to support roaming capabilities. Improving SCHC flexibility, scalability and assisting SCHC when a device is roaming are key considerations to increase the technology's adoption, and DNS might help by hosting rules outside the scope of the ED's associated backend without hindering the transmissions. To assist with this problem, we deployed a dynamic context resolution architecture based on DNS for SCHC compression/decompression and studied the consequences of such mechanism on the system latency and other possible consequences on LoRaWAN communications.

For this experiment, we built a SCHC-enabled LoRaWAN infrastructure. SCHC was able to efficiently compress headers from the 44-octets IPv6+CoAP header down to a few octets, which leads to reduce header size up to a 22-factor and enable the use of IPv6 over networks that would be unable to support it, such as SigFox and its 29 octets frame or LoRaWAN which ranges from 59 octets to 250 octets. Using SCHC to send IPv6 packets over LPWAN is proven to be an efficient way to consider the scarcity of radio resources. SCHC is also able to work within a reliable timeframe. Querying time within a huge database was not studied and would need additional data regarding actual SCHC usage on LPWAN to provide interesting insight, but when working with a few devices, SCHC can decompress data consistently with a few microseconds; an operation that is almost transparent with regards to other mechanisms specific to LPWAN, such as LoRaWAN's reception delays.

By exploiting the DNS infrastructure, one can query the context signature within a satisfying timeframe (between 30 and 100 milliseconds) and fall back onto the associated context storage API within 650 ms. In a best-case scenario, the 650 ms delay would be furtherly reduced by caching; our Atlas probes measurements lead us to believe that using a DNS-only mechanism and building a context cache would lead to reducing the 650ms delay down to tens of milliseconds. These results concerning the DNS-only system and its performances are consistent with results measured in other studies. Such a mechanism does not hinder the communication as it is kept under the delay of the first reception windows and benefits from the information caching should the answer need a different SCHC rule. Should we need to respond

to the device within the first reception window, 350ms of data handling are left in the worst-case to enqueue the response onto the RG.

Further work regarding the SCHC protocol would require additional data from the actual use of the SCHC protocol. Studying SCHC uses would help to define the research direction by putting them into contrast with the production issues introduced by the protocol. Studying SCHC context rules outside their theoretical construction, but rather based on actual rules used in production, would help further identify eventual new constraints introduced by the protocol and define useful research direction. Improving the compression capabilities of LPWANs is a crucial concern for the technology. Reducing packet size reduces airtime, which is an efficient way to improve the scalability of LPWAN solutions. Another solution to reduce airtime of LPWANs transmissions is transmission minimization. Our approach with transmission minimization is complementary to compressing header using SCHC. SCHC relies on suppressing and optimizing predictable data within the transmission's header. But what if the actual relevant data from the payload is predictable. In this case, would it be possible to study transmission minimization paradigms in which compressing communications would rely on predicting a device's transmission payload and preventing its transmission if it is unnecessary?

We decide to further our approach regarding transmission efficiency by compressing data. The easiest way to reduce data transmission is to delete redundancies or to round them to near values. When working with sensors, data is often time-correlated. For example, the temperature may vary slowly. Recently, Neural network-based techniques entered the landscape of IoT data compression techniques. Data can be compressed by their regression curve inferred from a neural network. More complex prediction methods can also be used. Neural networks are known as universal function approximators with the capability to learn arbitrarily complex mappings, and in practice, show excellent performance in prediction tasks. Nevertheless, if the "classical" methods present efficient compression ratios, they do not avoid transmitting data. Actually, periodically a sensor senses data, may compress it and then send the compressed payload, but compressed data payload and its associated header are still sent.

New neural network-based techniques appeared, and they avoid sending data in situations where the prediction is good. A neural network-based predictor is implemented in the ED and also in the backend. If the sensed data is well predicted, no data is sent, and the backend uses the prediction. Otherwise, it is sent. We experimented around these approaches by testing them in real experiments. Thus an actual LSTM implementation was developed and embedded on sensors to back or disprove the results obtained through simulations by the scientific community.

Our experiment studied the parameters to deploy such a neural network-based approach by experimenting with various use-cases such as varying the transmission decision threshold, the size of the neural network and the number of digits necessary to encode the weights and the variables. These parameters lead us to study the subsequent compression ratio and error rate, the energy consumption of the algorithm, the effect of quantization.

We built an experimental testbed to check the capabilities of onboarding LSTM algorithm on-sensor to forecast data, achieve dual prediction, and eventually compress data traffic and save energy. A deep LSTM algorithm was developed and integrated into small, constrained hardware to obtain these results; its source code is accessible

following [343]. Our findings show that it can efficiently minimize traffic while preventing non-relevant transmissions to occur with a significant impact on energy consumption. The overall system shows no significant impact from varying the neural network size and studied the impact from the decision threshold on the compression ratio and the MAPE. Our system allows efficient compression while keeping the user within a reasonable error margin. It can be customized depending on precision and compression trade-off requirements. The impact of the quantization of the LSTM parameters is checked because of device constraints and also to decrease the algorithm's complexity. No significant degradations in the system are observed when using 8-bit quantization. Our experiment shows that these machine learning algorithms can be easily embarked onto EDs, their performances are not lacking, and their storage size and energy consumption does not hinder the device's usual functioning.

Compressing data with this kind of Bounded Lossy Compression allows expanding battery lifetime depending on the accepted margin of error. Our results show an excellent compression ratio compared to the state-of-the-art. It is to note that our scheme avoids sending any data, while classical compression mechanisms at least send a frame header every time compressed data is sent. Moreover, these two approaches are complementary, and data can be compressed using classical compression mechanisms once the irrelevant information was suppressed. An extensive energy consumption study proved that our algorithm saves a portion of the device's energy that can be used in further communication. We developed arguments regarding ML weights storage capabilities for IoT infrastructure backed by DNS approaches. Such a solution seem feasible with regards to the binary size of the parameter file, but would require experimental work to back our assumptions with numbers.

Further approach would consist of selecting multiple features on multiple values to attain more precision in the calculation with a more complex recalibration. Works are carried by contributing to the actual TensorFlow Lite community to propose a complete port of the LSTM libraries from the global TensorFlow project to the TensorFlow Lite for microcontrollers community. Other works include studying the maximum supported size for neural networks to further our knowledge of small size neural networks and their performances.

In a nutshell, backing the DNS as a core service for interconnecting networks, hosting communication protocol rules to enhance IoT solution architecture in our different experimentation showed relevant results. We worked on building a roaming, easy to use, federated infrastructure to interconnect LoRaWAN networks as a solution to improve interoperability between IoT infrastructure backed by the DNS. We developed improvements to the SCHC protocol by hosting rules on the DNS infrastructure and allowing backend elements to query a global DNS zone that hosts the rules IDs and version number. Finally, we developed a transmission minimization algorithm by embedding a machine learning algorithm based on the LSTM algorithm onto LPWANs sensors and studied its impact on the underlying data and infrastructure. The results presented in this thesis show that the DNS, despite being one of the oldest protocols used on the Internet, can propose relevant improvements to infrastructure deployments and accompany new IoT use cases. The latest and ongoing work from the DNS community might assist in securing the aforementioned applications, such as switching from classical DNS to its more secure, latest implementations, as well as confirm information authenticity or assist in service discovery

to support IoT applications.

Appendix A

List of publications and communications

A. Bernard, S. Balakrichenan, M. Marot and B. Ampeau, "DNS-based dynamic context resolution for SCHC," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6, doi: 10.1109/ICC40277.2020.9148910.

A. Bernard, A. Dridi, M. Marot, H. Afifi and S. Balakrichenan, "Embedding ML Algorithms onto LPWAN Sensors for Compressed Communications," 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2021, pp. 1539-1545, doi: 10.1109/PIMRC50174.2021.9569714.

S. Balakrichenan, A. Bernard, M. Marot, B. Ampeau. "IoTRoam: design and implementation of an open LoRaWAN roaming architecture". 2021 IEEE Global Communications Conference (GLOBECOM), Dec 2021, Madrid, Spain. ⟨hal-03100628v3⟩

A. Bernard, M. Laroui, M. Marot, S. Balakrichenan, H. Moungla, B. Ampeau, H. Afifi and M. Becker, "Prefetching of mobile devices information - a DNS perspective", ICC 2022 - 2022 IEEE International Conference on Communications (ICC), 2022 (Submitted)

Appendix B

Résumé en français de la thèse

This section presents a short summary of the thesis, translated in French.

B.1 Introduction

L'Internet des Objets (IdO ou Internet of Things, IoT) a évolué depuis cette possibilité théorique de connecter tous les appareils à un réel marché de biens et de services en constante expansion. Les technologies sous-jacentes ont évolué et l'IoT repose aujourd'hui sur de nombreuses technologies de communication différentes: Des technologies à courte portée comme Bluetooth, moyenne portée comme Zigbee ou longue portée comme la technologie LoRa (Long-Range).

Les systèmes de l'IoT sont habituellement construits autour d'infrastructures fermées basées sur des systèmes en silo. Créer de l'interopérabilité entre ces silos fermés est un enjeu pour certains cas d'usages cruciaux dans le déploiement des technologies de l'IoT comme les villes intelligentes. Développer la problématique au niveau applicatif est une première étape directement inspirée des pratiques courantes en matière de collecte et d'analyse de données dans le cadre du développement des technologies de traitement de données massives. Cependant, construire des ponts au niveau réseau permettrait de faciliter l'interconnexion entre infrastructures et faciliterait la transition fluide entre technologies de l'IoT afin d'améliorer à bas coût la couverture réseau.

Le Système de Nom de Domaine (Domain Name System, DNS), initialement développé pour traduire les noms, lisibles et compréhensibles par les utilisateurs en adresses IP, utilisées par les appareils connectés, est reconnu comme un facilitateur sur les question d'interopérabilité sur Internet. C'est l'un des systèmes les plus anciens déployés sur Internet, développé à la fin des années 1980 pour supporter la croissance des infrastructures d'Internet. Bien qu'ayant beaucoup évolué ces dernières années, en témoignent les nombreuses propositions de modifications au standard publié à son sujet, le DNS reste aujourd'hui l'une des infrastructures les plus centrales du réseau Internet.

Le DNS repose sur des principes simples, mais son évolution depuis ses premiers développements ont permis de construire des systèmes complexes grâce à ses nombreuses possibilités de configuration. Dans le cadre cette thèse, qui étudie les possibles améliorations aux services et infrastructures de l'IoT, nous étudions la problématique suivante : Le DNS et son infrastructure peuvent-ils servir de support efficace à l'évolution de l'IoT de la même manière qu'il a accompagné l'évolution d'Internet ?

Ce manuscrit présente les travaux réalisés dans le cadre de la thèse de doctorat **Contributions à la résolution de problèmes de performances et d'interopérabilité des réseaux IoT hétérogènes par l'utilisation du standard ouvert DNS et de services d'infrastructure**. Cette thèse est réalisée en collaboration entre Télécom SudParis et l'Association Française pour le Nommage Internet en Coopération (AFNIC) dans le cadre d'une convention CIFRE. La thèse vise à proposer de nouvelles utilisations de l'infrastructures DNS permettant à celle-ci de servir de support aux diverses évolutions de l'Internet des Objets et ses systèmes. L'infrastructure DNS étant une infrastructure répartie déployée partout dans le monde, résiliente, supportée par la totalité des systèmes connectés à Internet et permettant aux utilisateurs d'Internet un accès facilité à un ensemble de services, exploiter une telle infrastructure plutôt que de redéployer des systèmes dédiés à l'Internet des Objets semble être une solution intéressante à considérer pour des usages étendus.

Dans cette optique, nous étudions de possibles améliorations de systèmes de l'IoT sous trois angles. Nous testons tout d'abord un modèle d'itinérance pour réseaux de l'Internet des Objets au travers de la construction d'une fédération reposant sur l'infrastructure du DNS et ses extensions pour en assurer l'interopérabilité, la sécurité de bout-en-bout et optimiser les communications entre infrastructures. Son objectif est de proposer des transitions fluides entre réseaux sur base d'informations stockées à l'aide de l'infrastructure DNS. Nous explorons également les problématiques introduites par le DNS, notamment en termes de latence et d'influence sur les temps de réponse des applications, et comment en limiter l'impact sur les échanges, déjà grandement contraints, entre objet connecté et passerelle radio. Pour cela nous étudions les conséquences de l'utilisation de requêtes DNS anticipées dans un contexte de mobilité en milieu urbain. Nous étudions ensuite la façon dont le Système de Nom de Domaine peut renforcer l'interopérabilité, la disponibilité de ressources et le passage à l'échelle de systèmes de compression de paquets de l'IoT. Enfin, nous explorons la question de la minimisation de trafic en implantant des algorithmes d'apprentissage sur des capteurs et en mesurant les paramètres du système final, en particulier en terme de performances de transmissions et d'efficacité énergétique.

B.2 IoTRoam, une fédération supportant l'itinérance pour l'IoT

Contrairement aux autres technologies IoT, LoRaWAN permet une diversité dans les stratégies de déploiement de réseau. On y retrouve des réseaux public, privés ou communautaire. Une infrastructure LoRaWAN publique se comporte de la même manière qu'un réseau cellulaire d'opérateur. En France certains opérateurs comme Orange ou Bouygues Télécom déploient d'ailleurs de tels réseaux. Un déploiement LoRaWAN privé est en général réalisé pour un cas d'usage précis par une institution - entreprise, hôpital ou université - qui ne déploie pas autant d'antennes qu'un opérateur de télécommunications et n'a besoin que d'une couverture limitée. Une solution communautaire, comme TheThingsNetwork, construit un gigantesque réseau autour d'une unique plateforme LoRaWAN qui peut être utilisée de manière indiscriminée et souvent gratuite par tous les membres de la communauté. Dans ce dernier cas la couverture géographique est limitée aux besoins spécifiques de la communauté.

L'itinérance dans les réseaux LoRaWAN peut être permise lorsqu'une interconnexion entre deux réseaux (public, privés ou communautaires) opérés par des institutions différentes est réalisée, l'itinérance permet alors à un appareil de continuer à transmettre et à recevoir des données même lorsque sa couverture réseau change et qu'il s'éloigne de son réseau d'origine. Pour l'instant, les déploiements d'infrastructures supportant l'itinérance dans le cadre des réseaux LoRaWAN ont été réalisés à l'aide de mécanismes tiers hors du canal de transmission, soit par l'établissement de connexions point-à-point soit à l'aide d'un mécanisme de Roaming Hub comme le propose Packet Broker [348]. Le Roaming Hub simule un système pair-à-pair en autorisant les partenaires à échanger des messages les uns avec les autres en passant par un point d'échange central commun.

Un problème persiste avec l'itinérance hors du canal de transmission : La solution hors-canal est-elle capable de passer à l'échelle considérant l'évolution croissante des déploiement de réseaux LoRaWAN ? La réponse serait affirmative si LoRaWAN ne disposait que de réseaux publics suffisamment peu nombreux pour définir ensemble des règles de coopérations au même titre que les opérateurs de téléphonie mobile au sein d'un pays. Cependant, une récente analyse du trafic actuel lié aux infrastructures LoRaWAN estime que 84% des communications en 2019 ont été réalisées dans un contexte de réseau privé ou communautaire contre 16% sur des réseaux administrés par un opérateur LoRaWAN public. N'utiliser que des mécanismes hors-canal pour établir des connexions entre une majorité (qui peut se chiffrer en milliers) de réseaux privés ou communautaires et une minorité de réseaux publics serait un cauchemar à administrer.

Dans le cadre des spécifications des infrastructures LoRaWAN, l'Alliance LoRa a alloué une zone DNS spécifique pour rassembler les adresses IP des réseaux LoRaWAN et les faire correspondre à leur adresse de réseau LoRaWAN, appelée NetID. Tout opérateur de réseau LoRaWAN (public, privé ou communautaire) peut demander à l'Alliance LoRa de se voir allouer un tel identifiant et renseigner dans la zone "netids.lorawan.net" les informations permettant de contacter son serveur réseau. Les spécifications proposent de faire correspondre le NetID et l'adresse d'un réseau se présente comme suit :

60050a.netids.lorawan.net. IN A 192.168.1.2

(60050a est le NetID et 192.0.2.45 l'adresse du Home Network Server)

Ainsi, tout serveur réseau LoRaWAN est en mesure d'identifier et de contacter le réseau d'origine de tout appareil transitant sur son réseau en interrogeant la base DNS, même sans accord d'itinérance préalable. Pour les appareils disposant d'accord d'itinérance entre LoRaWAN et d'autres réseaux (LTE-M, NB-IoT...), une fois que l'objet est hors de sa zone de couverture, il est également possible de retrouver l'adresse du réseau d'origine à l'aide d'une requête DNS.

Le DNS permet aussi de s'assurer qu'un NetID alloué à un opérateur réseau dans le cadre de l'Alliance LoRa, n'est pas dupliqué et utilisé par les autres et autorise les opérateurs de réseaux LoRaWAN à faire correspondre de manière dynamique le réseau de rattachement d'un objet connecté. Bien que l'utilisation du DNS soit à ce jour la seule solution proposée par l'Alliance LoRa dans le cadre des spécifications des infrastructures LoRaWAN pour supporter l'itinérance, l'Alliance ne met pas à disposition une telle infrastructure DNS à l'heure actuelle.

Notre travail vise à mettre en œuvre les spécifications LoRaWAN qui supporte l'itinérance, et à proposer la construction d'un réseau fédéré supportant le mécanisme d'itinérance entre infrastructures. Nous nous appuyons pour cela sur le système DNS. Notre cas d'usage propose d'étudier la faisabilité d'un déploiement d'infrastructures LoRaWAN entre universités partenaires, qui fonctionnerait de manière similaire à eduroam [306]. Le but d'une telle démarche est d'utiliser les fonctionnalités de roaming entre les partenaires du projet tout en leur permettant de rester maître de leur réseau, nous avons nommé ce projet IoTRoam. Nous mettons également en place une infrastructure à clefs publiques capable de générer et partager les certificats nécessaires à la communication, chiffrée à l'aide du protocole TLS, entre les partenaires du projet

Nous avons méticuleusement étudié les spécifications LoRaWAN et le code source disponible sur GitHub afin d'inclure la résolution DNS dans ChirpStack

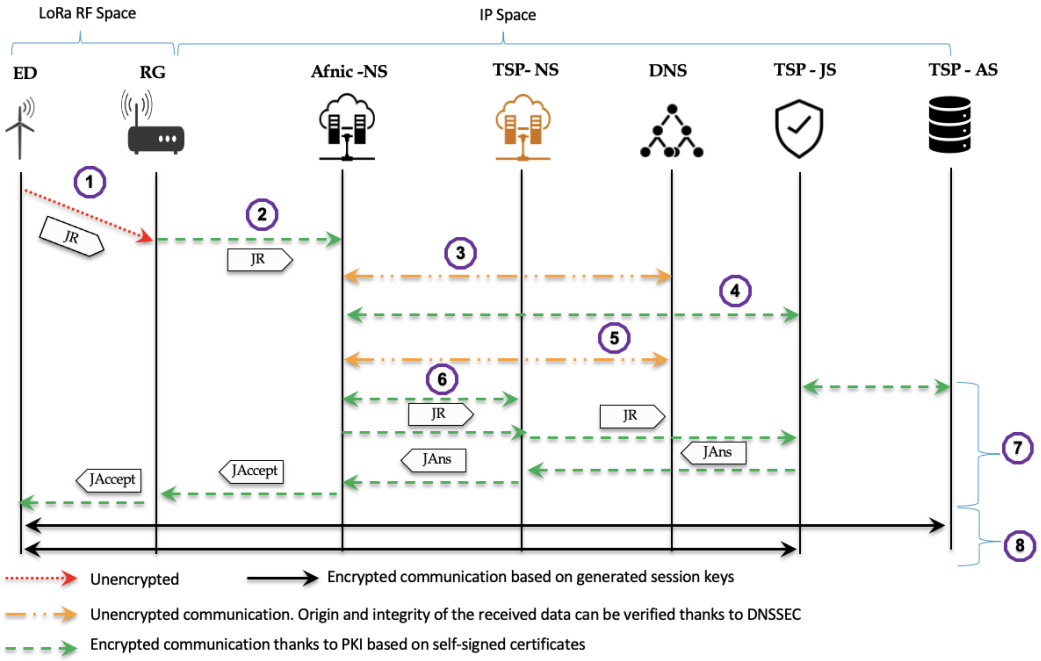


FIGURE B.1: Exemple d'utilisation du DNS comme support à l'OTAA

Nous avons ensuite testé le mécanisme d'activation "Over-the-Air" (OTAA - Figure B.1) dans un contexte Passive Roaming entre Télécom SudParis et l'AFNIC et documenté [349] le scénario , l'architecture et les cas étudiés afin de les partager avec les institutions partenaires.

Dans le scénario ci-dessus, nous avons utilisé une infrastructure PKI basée sur des certificats générés par une unique Autorité de Certification (CA). Nous avons mis en place une infrastructure pour laquelle chaque institution dispose de son propre certificat racine (que l'on appelle Intermediate CA) B.2 et que chacune délivre les certificats nécessaires aux communications avec ses machines à l'aide de ce certificat racine intermédiaire.

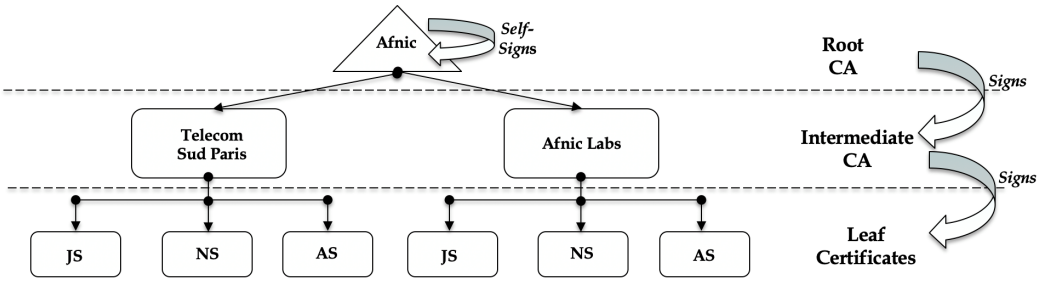


FIGURE B.2: Chaîne de confiance dans l'approvisionnement des certificats

Dans la suite des tests de notre infrastructure, nous avons évalué les performances du système d'itinérance en étudiant les conséquences de l'utilisation du DNS sur le système de communication LoRaWAN.

Nous avons pour cela étudié trois scénarios pour nos mesures :

- Scénario 1: L'appareil est au sein de son réseau mais sans latence introduite par le DNS ou l'infrastructure à clef publique.
- Scénario 2: L'appareil est au sein de son réseau mais les adresses du serveur réseau et du serveur d'activation sont résolus par le DNS.
- Scénario 3: L'appareil est au sein d'un réseau visité, les adresses du serveur réseau et du serveur d'activation d'origine sont résolus par le DNS et l'infrastructure à clef publique est utilisée pour sécuriser le trafic.

Les courbes ci dessous (Figures B.3 et B.4) présentent la répartition cumulées des temps d'activation et des délais aller-retour de communication pour nos appareils, réalisés directement sur ces derniers

Nos mesures nous laissent à penser qu'introduire le système DNS et une infrastructure à clef publique dans le fonctionnement de LoRaWAN n'ajoute pas de délais signifiants pouvant dégrader les communications. Notre système est configuré pour vider le cache DNS entre chaque requête, aussi un infrastructure réelle profiterait des bénéfices de celui-ci. Pour compléter cette étude nous avons également étudié les conséquence du pré-chargement d'informations à l'aide du DNS pour les appareils mobiles.

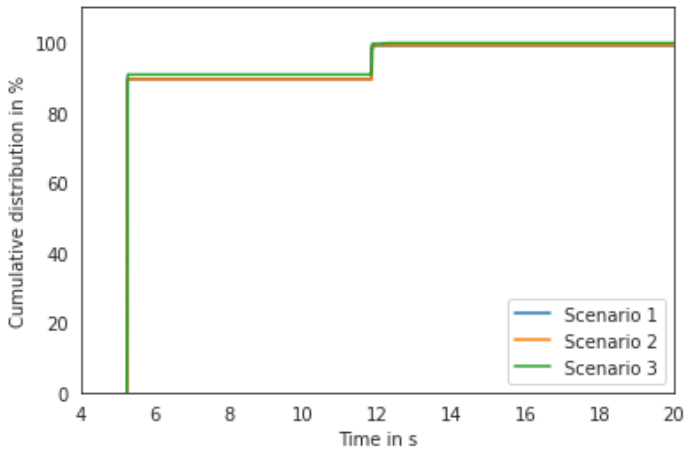


FIGURE B.3: Répartition (cumulée) des temps d'activation des appareils suivant nos trois scénarios

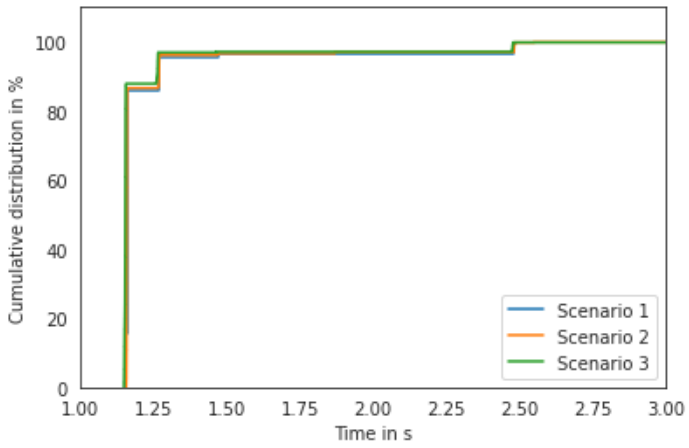


FIGURE B.4: Répartition (cumulée) des délai de communication entre appareil et infrastructure suivant nos trois scénarios

Pré-chargement d'informations de connexion des appareils mobiles pour réduire l'impact du DNS

Pour compléter notre étude sur les possibilité de monter une fédération d'itinérance sur un réseau LoRaWAN, nous avons étudié une possible méthode pour réduire l'impact réseau du mécanisme que nous introduisons dans les procédure de connexion. Dans un contexte de roaming, il est important pour les opérateurs de prioriser leurs connexions entrantes et de servir leurs utilisateurs le plus rapidement possible afin de ne pas surcharger leur propre réseau.

Le pré-chargement d'information est une stratégie classique pour réduire les délais de connexion, elle est utilisée principalement dans le contexte de la navigation web pour résoudre les domaines présents sur une page.

Le mécanisme de prédiction est assez simple, le lien étant présent sur la page, l'utilisateur peut être amené à cliquer dessus et donc l'information est correspondante est pré-chargée en prévision d'une éventuelle demande de la ressource par l'utilisateur. Un bon tutoriel sur le fonctionnement du pré-chargement est disponible sur le site du projet Chromium ([321]).

Nous émettons l'hypothèse dans ce travail qu'il serait possible d'exploiter les mécanismes de base du pré-chargement d'information DNS pour récupérer les informations de connexion au niveau de serveurs embarqués dans des antennes.

La méthode de prévision associée peut être basée sur du provisionnement de proximité, ou alors sur de la prédiction de trajectoire. Nous étudions ces deux approches afin de comparer leurs forces et leurs faiblesses.

Nos scénarios IoTRoam introduisent deux requêtes DNS dans le processus de négociation de canal entre appareil mobile et serveur. Notre questionnement pour cette partie est le suivant : "Est-il possible de réduire le délai introduit par nos requêtes DNS dans un contexte de mobilité ?"

Nous basons nos travaux sur des traces de mobilité de véhicules dans la ville de Rome. La figure B.5 illustre les traces étudiées en fonction de la position des véhicules (latitude et longitude).

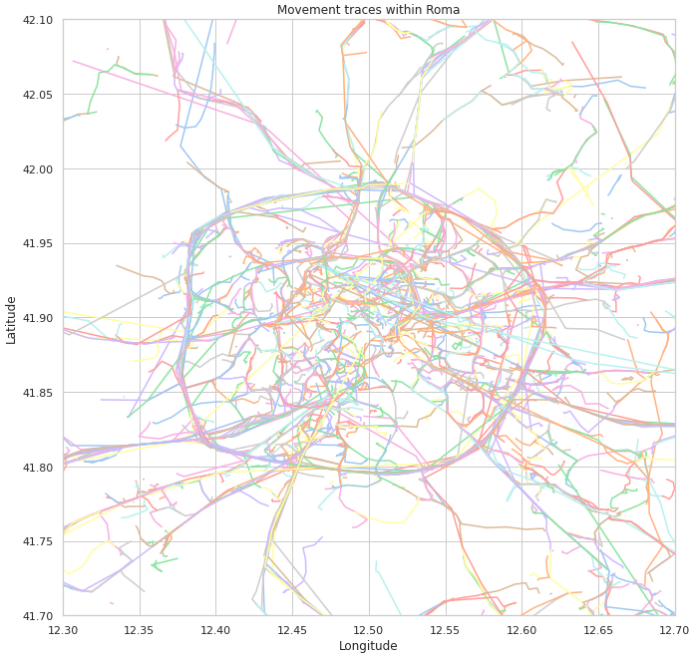


FIGURE B.5: Déplacements de véhicules dans la ville de Rome

À l'aide de simulations, nous quadrillons la ville de Rome à l'aide d'antennes. Nous simulons un déploiment LoRaWAN en mode "fog", chaque antenne est liée à un mini serveur, agit indépendamment et fournit aux appareils mobile une connexion itinérante.

Nous divisons notre étude en trois scénarios. Le premier fonctionne sans pré-chargement d'information, sur la base de requêtes DNS standards réalisées au cours de la tentative de connexion. Le second scénario améliore ce mécanisme en réalisant un pré-chargement sur les antennes les plus proches. Enfin, le troisième scénario introduit un mécanisme de prédiction de trajectoires, basé sur des algorithmes d'apprentissage, qui vient supporter le pré-chargement des données.

Pour les deux premiers scénarios, nous étudions la différence entre requêtes DNS et sollicitation du cache DNS. Pour le troisième, nous étudions plus en détail la manière dont le cache a été provisionné sur base des prédictions.

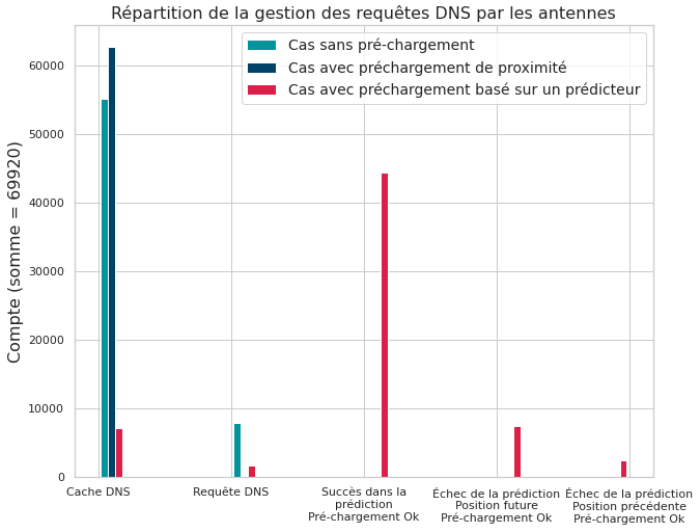


FIGURE B.6: répartition des sollicitations des caches entre les requêtes transférées aux antennes pour les différents scénarios étudiés

Les résultats, présentés sur la figure B.6 sont satisfaisant. Une prédiction est réalisée avec succès dans 70.4% des cas. Les requêtes au cache sur base d'échec de prédiction (décalage temporel dans la position prédite) liées à notre mécanisme permettrait également de supporter jusqu'à 86% des requêtes. Les 14% restants sont répartis entre les requêtes au cache classique (11.4%) et des requêtes DNS complètes sans pré-chargement (2.5%)

Nous avons également étudié la répartition de la sollicitation des antennes (Figure B.7). Nous constatons que le prédicteur entraine une sollicitation d'antennes moins importante que la distribution de proximité, mais plus que de ne pas réaliser de pré-chargement.

Nous en concluons que le pré-chargement d'information à l'aide du DNS constituerait une solution efficace capable de réduire les temps d'activation dans 86% des situations rencontrées, il se trouve moins efficace que le système basé sur les antennes de proximité mais plus qu'un système sans pré-chargement. Au niveau du nombre d'antennes sollicité, le système basé sur le prédicteur de trajectoires est cependant capable de réduire la charge en sollicitant moins d'antennes que le prédicteur de proximité.

Des études additionnelles sur base d'autres positionnement d'antennes, étudiant leur montée en charge ou étudiant d'autres traces de mobilité seraient intéressantes pour compléter ce travail. Une telle application du pré-chargement d'informations DNS est intéressante également dans le contexte de la récupération d'autres informations comme celles présentées en B.3 ou B.4.

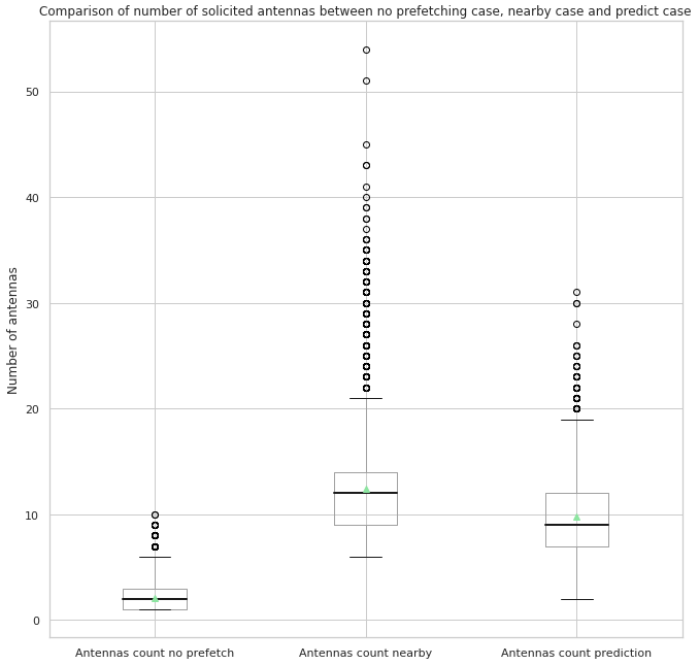


FIGURE B.7: Répartition des activations d'antennes pour nos trois scénarios

B.3 Système de résolution de Contexte pour le protocole SCHC à l'aide du protocole DNS

Le pré-chargement d'informations issues du DNS serait complémentaire à la proposition d'amélioration de SCHC [3] (RFC 8724 : Generic Framework for Static Context Header Compression and Fragmentation) que nous avons proposé dans le cadre de nos travaux. Il s'agit de profiter de la disponibilité et du passage à l'échelle du DNS pour délocaliser les règles de compression et de fragmentation. En effet, cela permet de réduire la charge côté réseau en évitant le stockage de toutes les règles de compression existantes, introduit de la souplesse dans l'utilisation de SCHC et facilite la mobilité des capteurs/actionneurs. Nous avons modifié la mise en œuvre OpenSource du protocole, développée par les personnes écrivant le standard, pour l'intégrer à une base DNS et avons mesuré le surcoût en termes de performances qu'une telle solution engendrerait.

SCHC permet de réduire la taille des paquets MAC en compressant les en-têtes des couches supérieures. SCHC est présenté par ses auteurs comme une solution, un framework, permettant de réduire efficacement la taille des en-tête IPv6 et UDP, bien que le standard prévoit d'évoluer pour traiter également des couches supérieures (notamment pour compresser les en-têtes CoAP). L'objectif de la solution SCHC tel qu'explicité par [350] est de fournir une surcouche protocolaire permettant de s'affranchir des spécificités de la technologie LPWAN sous-jacente. Cette surcouche adaptative peut également être fournie par un organisme tiers, ainsi [351] fournit-il aux développeurs souhaitant utiliser des messages DLMS sur des réseaux LPWAN un cadre de développement leur permettant d'utiliser SCHC et d'économiser du temps de développement dans leurs applications tout en maintenant le niveau de sécurité nécessaire dans le cadre de leurs communications. SCHC fournit une méthode efficace de réduction de la taille de l'en-tête des paquets IPv6, d'une taille fixe

de 40 octets, en la faisant correspondre à une en-tête plus adaptée à la technologie LPWAN d'environ 2 octets. À titre indicatif, les paquets LoRaWAN font entre 51 et 222 octets selon le facteur d'étalement (et donc, la distance de transmission) choisi, les paquets SigFox 12 octets. Quant aux technologies LTE-M et NB-IoT, bien que supportant des tailles de paquets plus élevées, les performances sont supérieures sur des paquets de taille réduite, par exemple [352] relève des débits deux fois plus importantes chez LTE-M pour des paquets de 30 octets) que pour des paquets de 200 octets ou plus. SCHC fournit également un framework permettant de fragmenter les paquets et de les réassembler aisément à la destination. Une telle solution permet d'envoyer des paquets IPv6 complets sur les réseaux LPWANs, ce qui, au vu des tailles de paquets sur de tels réseaux, n'est pas possible.

Notre travail fut de proposer de faire évoluer SCHC afin de permettre aux utilisateurs de mettre les règles qu'ils veulent pour la compression dans une base de données DNS. En effet, le standard et les mises en œuvre OpenSource actuelles de SCHC proposent uniquement un stockage statique des règles de transmission. Or il faut stocker l'ensemble des règles sur l'infrastructure réseaux puisque, SCHC étant utilisé sur la voie radio entre le capteur/actionneur et l'antenne, il faut stocker sur le réseau (antenne ou serveurs) l'ensemble des règles de tous les capteurs susceptibles d'arriver sous sa couverture. Stocker en l'état les données nécessaires à la transmission utilisant la technologie SCHC au niveau de l'infrastructure réseau semble facile, les règles de transmission ne prenant que quelques Mo. Cependant dans un contexte d'usage massif des technologies IoT (environ 5000 capteurs autour d'une antenne), constituer une base de donnée de l'ensemble des règles pour les capteurs autour d'une seule antenne demanderait, pour les seules opérations liées à SCHC, de stocker quelques gigaoctets de données. Le stockage externe des règles SCHC nous semble, dans un tel contexte, devenir une nécessité. Cela permettrait un accès facilité aux règles et réduirait la pression qu'exerce SCHC au niveau des infrastructures réseau. Cette solution permettrait également aux utilisateurs de garder un œil sur les changement de profil d'appareil, de fournir des statistiques sur la mobilité des appareil tout en gardant une qualité de service acceptable vis à vis de l'ensemble des appareils contraints. Enfin, à partir du moment où les règles peuvent être trouvées facilement, en théorie n'importe quel capteur mobile peut être servi par n'importe quelle antenne.

À notre connaissance, la question de la récupération sur un serveur distant des règles SCHC n'a été examinée que dans [4] qui propose d'utiliser un Administration Management Server (AMS) comme support à l'itinérance sur un réseau LoRaWAN et où seraient donc stockées les règles SCHC. Cependant la résolution permettant de récupérer l'adresse de l'AMS demanderait, dans un tel contexte, de nombreux accords d'itinérance entre les opérateurs déployant des réseaux LoRaWAN. Le réseau LoRaWAN étant basé sur une communauté ouverte et facile d'accès, une telle solution à base d'accords d'itinérance serait coûteuse à mettre en œuvre ou restreindrait drastiquement les réseaux en mesure de faire de l'itinérance. La solution basée sur DNS que nous proposons dans notre article nous permet de nous affranchir d'une telle complexité.

Au cours de nos expériences, nous avons transmis à l'aide des capteurs/actionneurs des données compressées. Au niveau de notre infrastructure LoRaWAN, nous avons extrait les données transmises et testé différents scénarios de décompression. Le premier scénario est avec compression statique, le second avec compression et règles délocalisées et le troisième avec compression et règles délocalisées et utilisation

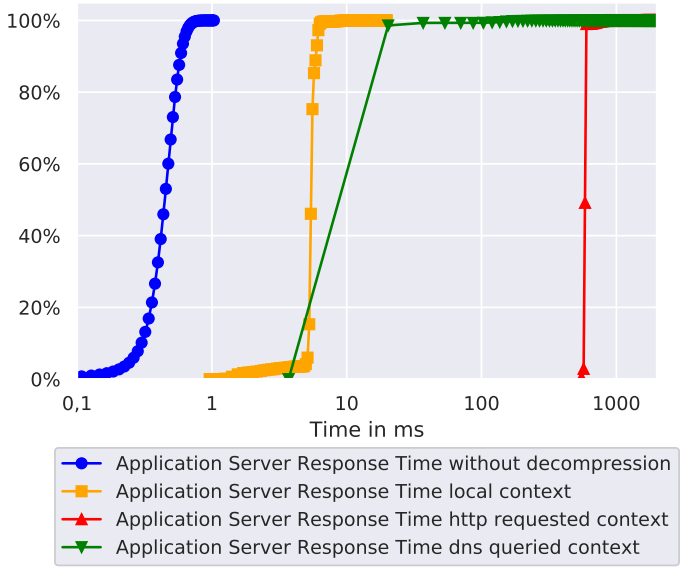


FIGURE B.8: Fonction de répartition des temps de réponse du système support à la compression en fonction des scénarios étudiés dans notre article

d'un cache pour le stockage des règles. Nous avons comparé les variations au niveau de la latence du système due à l'ajout dans le processus de décompression SCHC d'une requête DNS et d'une requête HTTP réalisée sur une API (Figure B.8). Cette comparaison montre que les délais ajoutés sont acceptables au regard des délais du protocole (par exemple les durées d'ouvertures des fenêtres illustrées figure B.9). En utilisant le système DNS, un système déjà utilisé par de nombreux développeurs et accessible mondialement, nous avons proposé un système de récupération distant de règles SCHC permettant aux infrastructures LoRaWAN de récupérer de manière dynamique les règles de compression et fragmentation SCHC. Ce mécanisme ajoute, certes, un délai dans la communication dû au mécanisme de résolution DNS mais ce délai est parfaitement acceptable si l'on prend en considération les règles de transmission propres aux contraintes des appareils de classe A (Figure B.9) que toutes les solutions à base de LoRaWAN sont obligées de supporter.

Cette étude a également permis d'étudier la latence induite par la compression SCHC sur un système, en réalisant des mesures comparatives entre notre premier scénario à compression statique et un scénario de test sans compression, ce qui n'avait jamais été réalisé auparavant.

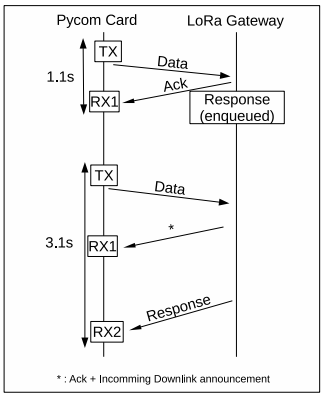


FIGURE B.9: Contraintes de transmission LoRa pour les appareils de classe A

B.4 Réduction de trafic réseau assisté par Apprentissage Machine

Après avoir exploré la question de la compression d'en-tête, nous avons décidé de regarder celle de la compression de la charge utile de nos paquets. De nouveaux algorithmes de compression basés sur l'apprentissage machine voient le jour. Des solutions reposent sur des techniques de classification comme [353] ou ce que propose EdgeImpulse[339], qui fournit un framework permettant d'embarquer un classificateur capable de traiter, par exemple, un signal audio, ou de deviner un mouvement et d'envoyer l'information du changement de mouvement sur voie radio. D'autres reposent sur des prédicteurs de séries temporelles (cf. [158] et [335]). Le principe est d'avoir un prédicteur de la donnée, sous forme de réseau de neurones du type Long Short-Term Memory (LSTM) B.10, d'exécuter ce prédicteur sur le capteur et côté infrastructure et, lorsqu'une donnée captée doit être émise, de vérifier si elle coïncide, à un seuil de décision près, avec la donnée prédite à ce moment. Si c'est le cas, la donnée n'est pas émise. Elle l'est sinon. Comme on peut attendre des taux de prédiction corrects de l'ordre de 90%, cela permet d'espérer des ratios de compression du même ordre. Ceci revient à isoler les données exceptionnelles (pics de consommation, changements d'amplitude de mouvement...) et à prendre la décision de ne transmettre que dans ces cas précis. Plus précisément, nous nous sommes basés sur [335]. Si cette méthode fonctionne en théorie, il reste de nombreuses questions en pratique. D'une part elle n'a jamais été testée dans un environnement réel. D'autre part, elle peut causer des difficultés de mise en œuvre dues à la capacité restreinte des équipements en termes de capacité CPU et mémoire, ou en termes de consommation énergétique mais aussi poser des problèmes réseaux. En effet, si la compression est parfaite, la liaison n'est plus maintenue et il faut donc envoyer artificiellement des paquets keep-alive pour maintenir la connexion: d'un côté on diminue la charge de trafic mais de l'autre on en rajoute. Une validation s'impose donc en environnement réel.

L'utilisation de tels modèles nous empêche de faire usage de la solution EdgeImpulse, qui nous aurait été bien utile dans notre travail. En effet, EdgeImpulse, de part son architecture, ne permet de faire que de la classification de données et ne nous permet pas de personnaliser suffisamment nos outils afin de réaliser nos expérience. Nous avons donc embarqué TensorFlow Lite sur des cartes de développement

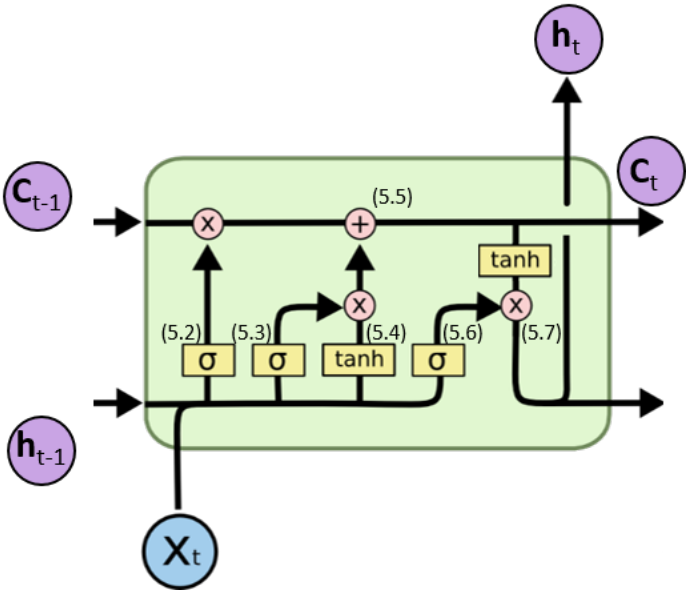


FIGURE B.10: Réseau de Neurone LSTM (extrait de [340])

STM32 en le compilant depuis ses sources et en y ajoutant une solution pour transmettre sur un réseau LoRaWAN à l'aide de notre carte de développement. La solution TensorFlow Lite nous permet de construire un réseau de neurones pré-entrainé à l'aide de TensorFlow, puis d'extraire les poids nécessaires à son fonctionnement et d'embarquer ces derniers dans un appareils contraint. Cependant, le réseau de neurone, LSTM B.10, mis en œuvre dans le cadre de la génération de séries temporelles n'est à ce jour pas supporté par TensorFlow Lite.

Une fois que nous avons mis en œuvre le réseau LSTM, nous avons embarqué notre algorithme sur des cartes électroniques et mesuré leur consommation énergétique ainsi que divers paramètres au niveau logiciel (nombre de neurones dans le réseau, performances de compression, taux d'erreur, seuil de transmission et quantification des poids).

Les courbes ci-dessous représentent les différentes mesures effectuées pour ce travail/ La figure B.11 présente la consommation énergétique de notre carte électronique. On y observe des différences de consommation marquées entre repos et activité. On constate que lorsque nos opérations sont réalisées, on peut observer les pics d'activités classiques pour la carte de transmission, ou aucun pic, correspondant à une transmission évitée par notre algorithme. Dans le cas usuel, nous observerions des pics de consommations réguliers pour nos deux cartes mais avec notre méthodes, ces pics sont complètement supprimés.

Les figures B.12a et B.12b présentent le MAPE (Mean Absolute Percentage Error, Équation B.1) et le taux de compression observé pour nos algorithme selon différents scénarios (variation dans le nombre de neurones ou dans le jeu de données étudié). Nous n'observons pas de différence significative en fonction de la taille des réseaux de neurones. Par contre, la figure B.12b nous montre l'importance du choix du jeu de données dans la performance de l'algorithme de prédiction, et donc dans les conséquences de la compression et du taux d'erreur observé.

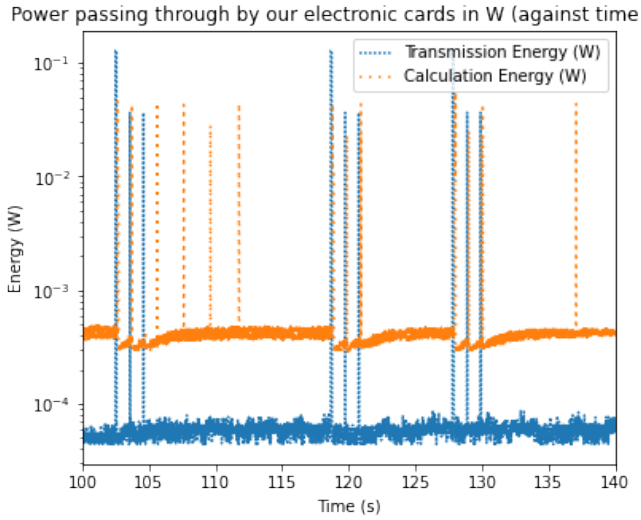
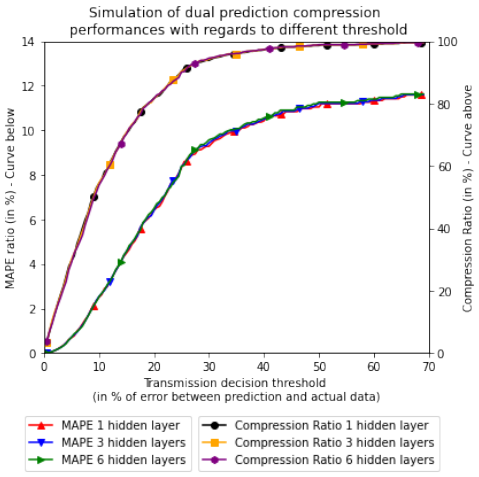
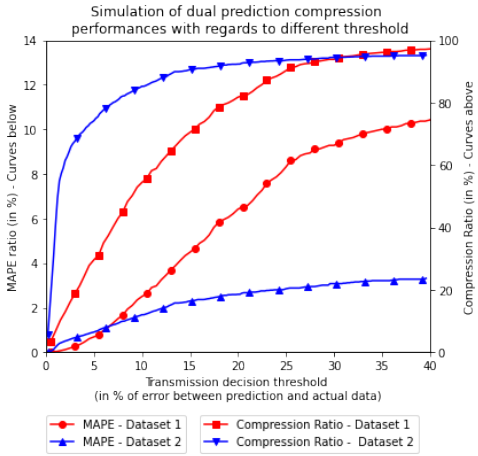


FIGURE B.11: Énergie (in W) alimentant la carte de transmission et la carte de calcul (échantillon en fonction du temps)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{C_i - M_i}{C_i}\right| \tag{B.1}$$



(A) Taux de compression et pourcentage d'erreur moyen en fonction de la taille du réseau de neurone et du seuil de précision choisi



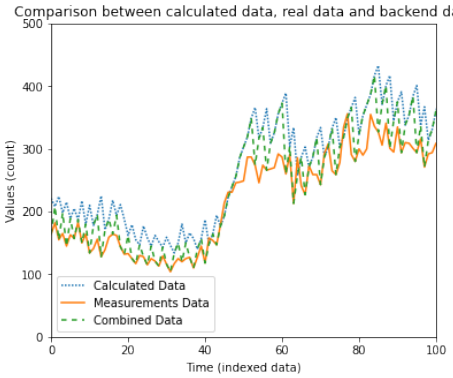
(B) Taux de compression et pourcentage d'erreur moyen en fonction du seuil de précision choisi pour différents jeux de données

FIGURE B.12: Taux de compression et pourcentage d'erreur moyen selon différents seuils de prédiction

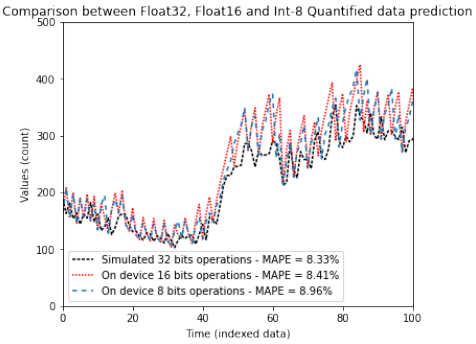
Les figures suivantes (B.13a et B.13b) présentent un échantillon des données afin d'illustrer un comparatif entre données ainsi que les conséquences de la quantification.

La figure B.13a est illustrative, elle présente la courbe finale consolidée vue au niveau du serveur déterminée en combinant les données calculées par le serveur et les données transmises par le capteur.

La figure B.13b quant à elle présente les conséquences de la quantification. La quantification sur 8-bits est un système connu et bien documenté permettant de réduire efficacement la taille des réseaux de neurones et des opérations effectuées par le capteur. Nous observons une faible dégradation des performances du système mais celle-ci reste dans des marges raisonables. Réduire la complexité des opérations permet de réduire les besoins en énergie et en puissance de calcul au niveau du capteur. Aussi, continuer les travaux sur la quantification nous semble intéressant une fois que l'algorithme est officiellement porté dans la librairie TensorFlow Lite.



(A) Échantillon comparatif entre données calculées, données de référence et données perçues par le serveur de réception



(B) Prédiction de données à l'aide de poids quantifiés sur 32 bits (Float32), 16 bits (Float16) et 8 bits (Int-8)

FIGURE B.13: Échantillon de données étudiées analysées (comparatif illustratif et quantification)

Dans le cadre de cette expérience, nous nous demandons également si les poids permettant la compression à l'aide d'un réseau de neurones peuvent être ajoutés au DNS afin qu'ils soient résolus par les infrastructures et qu'ils puissent être facilement partagés dans un contexte, par exemple, de mobilité des capteurs. La solution TensorFlow Lite, permettant d'extraire toutes les informations nécessaires au fonctionnement d'un réseau de neurones en respectant les spécificités des objets contraints (notamment la taille des structures de données), semble fournir les outils adaptés à la réalisation d'une telle démarche.

Utiliser ainsi DNS serait aussi possible dans le cadre de modèles de réseaux de neurones basés sur la classification, comme ceux utilisés par EdgeImpulse, ceux-ci ayant un poids de quelques Ko. Cependant les modèles utilisés pour la prédiction étant plus complexes, les poids en résultant le sont également, l'extraction de ceux-ci nous laissant avec un fichier d'une centaine de Ko.

Stocker les poids de notre algorithme sur mesure serait cependant faisable. Le caractère statique de celui-ci enlevant une grande quantité d'information, assez lourde, des données exportées par TensorFlow Lite. Nous détaillons dans ce manuscrit différents scénario de stockage de poids de réseaux de neurones dans le DNS suivant 3 paradigmes : Usage classique du DNS, utilisation combinée avec un système de stockage plus conséquent ou exploitation des fonctionnalités DNS-SD en réseau "mesh".

Nous avons, par ce travail, été en mesure de construire un banc de test expérimental nous permettant de mesurer les performance de notre système de minimisation de trafic IoT pour les réseaux de capteurs. Nos mesures montrent que le système est capable de réduire de manière efficace le trafic observé, mais que cette efficacité est

relative au cas d'usage des utilisateurs. Nous avons étudié le sujet sous divers angles comme le nombre de neurones dans le réseau, les performances de compression, le taux d'erreur, le seuil de transmission et la quantification des poids du réseau.

Utiliser un tel system permettrait d'augmenter la durée de vie des batteries des capteurs. Il faut également noter que notre mécanisme supprime complètement le trafic réseau associé et peut donc être utilisé de manière complémentaire à d'autres mécanismes de compression basé sur l'analyse des données binaires transmises ou des en-têtes.

B.5 Conclusions

Cette thèse a présenté diverses approches pour améliorer l'interopérabilité et les performances des solutions IoT basées sur LoRaWAN en exploitant l'infrastructure DNS existante comme terrain d'entente pour les développeurs d'applications et les chercheurs. Le DNS est un protocole connu de la communauté Internet, avec diverses implémentations open-source pour les clients, les serveurs ou au sein de frameworks. Des outils ont été déployés pour mesurer ses performances et moduler son utilisation. Sa communauté est ouverte et propose de nouveaux cas d'usage et des améliorations qui permettent de maintenir la communauté DNS à jour des nouveaux développements, protocoles et paradigmes réseau développés par le monde industriel ou la communauté de recherche.

Au sein de l'écosystème IoT, LoRaWAN est flexible ; soutenu par une communauté toujours plus importante d'acteurs industriels et une communauté académique officielle nouvellement créée, l'Alliance LoRa et son protocole LoRaWAN se pose comme un acteur majeur de l'écosystème IoT. Les évolutions issues des discussions de l'Alliance LoRa sont suivies de près par les développeurs d'applications LoRaWAN, puisque la pile LoRaWAN open-source de référence, ChirpStack, a connu 17 versions mineures et 1 version majeure au cours des 3 dernières années.

Les applications IoT rencontrent, avec leur dernier développement, les mêmes problèmes que les applications Internet. Les technologies IoT sont confrontées à des problèmes d'évolutivité, d'interopérabilité, de mobilité et d'itinérance, d'efficacité de transmission, de disponibilité, de fiabilité et d'autres problèmes de sécurité tels que la confiance et la confidentialité. Le DNS contribue à résoudre nombre de ces problèmes sur Internet, d'où notre interrogation sur les améliorations possibles des systèmes IoT soutenus par l'infrastructure DNS.

Cette thèse a étudié les systèmes IoT en ce qui concerne les aspects clés suivants : Le nommage, l'itinérance, la compression des en-têtes et la compression des données utiles. Cette étude ne visait pas à mettre en œuvre le protocole DNS sur les capteurs mais plutôt à utiliser le DNS du côté de l'infrastructure pour soutenir les améliorations de l'IoT. Cette thèse a présenté un travail expérimental sur LoRaWAN concernant divers scénarios pour tester des solutions, des applications et des cas d'utilisation IoT. Cette approche expérimentale a introduit des contraintes supplémentaires telles que travailler avec des implémentations de référence des solutions, générer un trafic IoT réel pour les mesures et l'analyse, respecter les contraintes de temps d'antenne ou le cycle de vie des appareils.

L'expérimentation de l'itinérance nécessite un accord d'interconnexion entre les opérateurs de réseau, généralement basé sur une interconnexion "One-to-One" ou par la construction d'un "Hub" d'interconnexion. Nous avons exploité une approche fédérée de l'interconnexion IoT en proposant l'architecture IoTRoam, fédérant différentes organisations pour permettre une authentification et une autorisation mutuelles flexibles entre n'importe quel élément de l'infrastructure dans des situations d'itinérance, sans accord d'itinérance direct et explicite (en interconnectant les serveurs de réseau, d'application et de jointure entre les opérateurs). L'accord d'interconnexion est implicitement donné lorsque l'organisation rejoint la fédération IoTRoam.

Notre architecture propose une solution qui prend en compte les contraintes caractéristique des environnements IoT. Notre approche pour construire notre architecture d'itinérance a été d'utiliser la combinaison de l'infrastructure DNS et d'une PKI pour construire une infrastructure d'itinérance ouverte sécurisée accessible aux opérateurs LoRaWAN publics et privés. Nous tirons parti de la possibilité de créer librement des réseaux LoRaWAN privés. Nous avons conçu, construit et déployé une architecture de preuve de concept pour tester les capacités d'itinérance offertes par la solution ChirpStack et tester l'itinérance entre les réseaux LoRaWAN privés et publics. L'infrastructure a été validée en testant la connectivité LoRaWAN pour les appareils dans un contexte d'itinérance en étudiant différents scénarios de connexion, en mesurant le temps d'établissement de connexion et les délais de communication.

Nous avons étudié les conséquences de la mise en cache et du pré-chargement d'informations DNS avec des appareils mobiles dans une ville par des simulations de communications entre l'appareil mobile et l'infrastructure IoT. Le pré-chargement DNS est un outil efficace pour réduire les requêtes DNS à la volée nécessaires à la communication entre les appareils. Le pré-chargement des informations sur les antennes proches peut complètement éviter les requêtes DNS en les exécutant à l'avance autour des antennes les plus proches, mais cela a un coût, car les appareils demandent plus d'antennes, surtout dans un environnement très mobile.

Notre combinaison d'un prédicteur ML et du pré-chargement permet une réduction intéressante des requêtes DNS réalisées par rapport à une solution standard de mise en cache uniquement et une réduction du nombre de passerelles réalisant l'opération de pré-chargement par rapport à la sollicitation des antennes les plus proches. L'utilisation du DNS nous permet d'exploiter son positionnement en tant que solution de base de données distribuée connue pour remplir des caches localisés afin de fournir des informations dès que nécessaire et de les purger au fil du temps lorsque les appareils mobiles quittent la couverture des antennes.

Nos simulations ont été réalisées dans le cadre de la fourniture d'informations de connectivité itinérante aux dispositifs, mais pourraient être applicables lors de l'interrogation d'autres informations nécessaires à la communication des dispositifs, comme les certificats stockés avec DANE, les paramètres de compression ou toute information spécifique au dispositif stockée dans le DNS.

Nous avons construit une infrastructure viable qui pourrait être intégrée dans les infrastructures IoT existantes avec un minimum de changements. Nous avons suivi les directives de la WBA pour l'itinérance ouverte et satisfait aux exigences décrites en employant des normes ouvertes utilisées sur Internet pour réaliser notre vision. Ces expériences ont conduit à trois "Change Requests" adoptées après avoir été soumises à l'Alliance LoRa.

Nous avons discuté de l'initiative IoTRoam avec plusieurs institutions afin d'effectuer des tests d'interopérabilité à l'aide de la plateforme fédérée ; l'exécution de tests supplémentaires avec ces institutions nous aiderait à étudier l'impact des infrastructures hétérogènes et leur effet sur la qualité du canal de communication et nous permettrait également de recueillir des données supplémentaires sur l'impact de la résolution complète du DNS sur le trafic LoRaWAN. L'objectif étant d'interconnecter des réseaux utilisant différentes technologies IoT, les prochaines étapes consistent à tester l'interopérabilité de l'itinérance avec NB-IoT, 5G ou Wi-Fi. Pour la connexion des appareils connectés, nous travaillons également à

l'intégration de DANE avec DNSSEC puisque les données du certificat lui-même peuvent être stockées dans le DNS, remplaçant ou complétant éventuellement la PKI.

En complément de notre travail d'interconnexion des infrastructures IoT, nous avons expérimenté la gestion et le partage des règles SCHC comme solution pour améliorer l'évolutivité des solutions LPWANs. Fournir un moyen d'échanger des informations sur les règles entre les infrastructures offrirait de nouvelles possibilités pour fournir plus de flexibilité au réseau, améliorant ainsi la flexibilité des solutions IoT dans des contextes tels que les dispositifs itinérants. Le mécanisme proposé exploite l'infrastructure DNS comme solution pour améliorer les capacités d'interrogation des règles au sein du réseau. Malheureusement, les règles sont trop lourdes pour être intégrées directement dans les enregistrements de ressources DNS ; un mécanisme de secours a donc été conçu sur la base des API et exploite le mécanisme de mise en cache du DNS pour stocker les identifiants et les numéros de version des règles. Le DNS, en tant que base de données optimisée, hiérarchique et distribuée, pourrait aider à identifier l'emplacement du serveur où les règles contextuelles sont stockées, et ce, de manière réaliste sur Internet. Il est à espérer que l'utilisation d'un tel mécanisme permettrait une transition transparente, de la préconfiguration des informations nécessaires en arrière-plan à leur construction dynamique, à la volée, sur la base des besoins réels lors de l'exploitation du réseau.

Le DNS s'avérerait une solution efficace pour introduire plus de flexibilité et améliorer l'évolutivité lors de l'utilisation du SCHC. Notre solution fournirait un accès ouvert aux paramètres de SCHC afin de soutenir les capacités d'itinérance. L'amélioration de la flexibilité et de l'évolutivité de SCHC lorsqu'un dispositif est itinérant sont des considérations essentielles pour accroître l'adoption de la technologie, et le DNS pourrait aider en hébergeant des règles en dehors du champ d'application de l'infrastructure associé à l'appareil connecté sans entraver les transmissions. Pour aider à résoudre ce problème, nous avons déployé une architecture de résolution de contexte dynamique basée sur le DNS pour la compression/décompression SCHC et avons étudié les conséquences d'un tel mécanisme sur la latence du système et d'autres conséquences possibles sur les communications LoRaWAN.

Pour cette expérience, nous avons construit une infrastructure LoRaWAN compatible avec SCHC. L'utilisation de SCHC pour envoyer des paquets IPv6 sur LPWAN s'avère être un moyen efficace de prendre en compte la rareté des ressources radio. SCHC est également capable de travailler dans un délai fiable. Les délais d'interrogation en base de données n'ont pas été étudiée et nécessiterait des données supplémentaires concernant l'utilisation réelle de SCHC sur les réseaux LPWAN pour fournir un aperçu intéressant, mais lorsqu'il travaille avec quelques appareils, SCHC peut décompresser les données de manière constante en quelques microsecondes ; une opération qui est presque transparente par rapport à d'autres mécanismes spécifiques aux LPWANs, tels que les délais de réception de LoRaWAN.

En exploitant l'infrastructure DNS, on peut interroger la signature du Contexte dans un délai satisfaisant (entre 30 et 100 millisecondes) et se rabattre sur l'API de stockage de Contexte associée dans un délai de 650 ms. Dans le meilleur des cas, le délai de 650 ms serait encore réduit par la mise en cache ; les mesures de nos sondes Atlas nous amènent à penser que l'utilisation d'un mécanisme exclusivement DNS et la construction d'un cache de contexte permettraient de réduire le délai de 650 ms à quelques dizaines de millisecondes. Ces résultats concernant le système DNS seul et

ses performances sont cohérents avec les résultats mesurés dans d'autres études. Un tel mécanisme n'entrave pas la communication car il est maintenu sous le délai des premières fenêtres de réception et bénéficie de la mise en cache des informations si la réponse nécessite une règle SCHC différente. Si nous devons répondre au dispositif dans la première fenêtre de réception, il reste 350 ms de traitement des données dans le pire des cas pour mettre la réponse en file d'attente sur la passerelle radio.

Des travaux supplémentaires concernant le protocole SCHC nécessiteraient des données additionnelles provenant de l'utilisation réelle du protocole SCHC. L'étude des utilisations de SCHC aiderait à définir l'orientation de la recherche en les mettant en contraste avec les problèmes de production introduits par le protocole. L'amélioration des capacités de compression des LPWANs est une préoccupation cruciale pour cette technologie. La réduction de la taille des paquets permet de réduire le temps d'occupation des antennes. Une autre solution pour réduire ces délais d'occupation est la minimisation des transmissions. Notre approche de la minimisation des transmissions est complémentaire aux autres méthodes de compression.

Nous décidons d'approfondir notre travail sur l'amélioration dans l'efficacité de transmission en compressant les données. La façon la plus simple de réduire la transmission des données est de supprimer les redondances ou de les arrondir à des valeurs proches. Or, lorsque l'on travaille avec des capteurs, les données sont souvent corrélées dans le temps ; par exemple, la température peut varier lentement. Récemment, les techniques basées sur les réseaux neuronaux ont été proposées pour compresser les données de l'IoT. Les données peuvent par exemple être compressées par leur courbe de régression déduite d'un réseau de neurone. Des méthodes de prédiction plus complexes peuvent également être utilisées. Les réseaux de neurone sont reconnus comme des approximateurs de fonctions universelles efficaces capables d'apprendre des motifs complexes et, en pratique, ils affichent d'excellentes performances dans les tâches de prédiction. Néanmoins, si les méthodes "classiques" présentent des taux de compression efficaces, elles n'évitent pas la transmission des données. En effet, un capteur capte périodiquement des données, peut les compresser puis envoyer la charge utile compressée, mais la charge utile compressée et son en-tête associée sont toujours envoyées.

De nouvelles techniques basées sur les réseaux neuronaux sont apparues, et elles évitent d'envoyer des données dans les situations où la prédiction est bonne. Un prédicteur basé sur un réseau de neurone est mis en œuvre dans l'appareil connecté et également dans l'infrastructure applicative. Si les données détectées sont bien prédites, aucune donnée n'est envoyée et l'application utilise la prédiction. Dans le cas contraire, les données sont envoyées. Nous avons expérimenté ces approches en les testant à l'aide d'expérimentations en situations réelles. Ainsi, une implémentation réelle de LSTM a été développée et embarquée sur des capteurs pour confirmer ou infirmer les résultats obtenus par les simulations de la communauté scientifique.

Notre expérience a étudié les paramètres de déploiement d'une telle approche basée sur un réseau de neurones en expérimentant divers cas d'utilisation tels que la variation du seuil de décision de transmission, la taille du réseau de neurones et le nombre de bits nécessaires pour encoder les poids et les variables. Ces paramètres nous amènent à étudier le taux de compression et le taux d'erreur, la consommation énergétique de l'algorithme, l'effet de la quantification.

Nous avons construit un banc d'essai expérimental pour vérifier les capacités de l'algorithme LSTM embarqué sur le capteur à prédire les données, réaliser une double prédiction, et finalement compresser le trafic de données et économiser de l'énergie. Un algorithme LSTM a été développé et intégré dans un appareil contraint pour obtenir ces résultats ; son code source est accessible à l'adresse suivante : [343]. Nos résultats montrent qu'il peut minimiser efficacement le trafic tout en empêchant les transmissions non pertinentes de se produire avec un impact significatif sur la consommation d'énergie. Le système global ne montre aucun impact significatif de la variation de la taille du réseau de neurone et nous avons pu étudier l'influence du seuil de décision sur le taux de compression et le MAPE. Notre système permet une compression efficace tout en maintenant une marge d'erreur raisonnable pour l'utilisateur. Il peut être personnalisé en fonction des exigences de précision et de compression. L'impact de la quantification sur les paramètres du LSTM a été étudié en raison des contraintes du dispositif et aussi pour diminuer la complexité de l'algorithme. Aucune dégradation significative du système n'est observée en utilisant une quantification 8-bits. Notre expérience montre que ces algorithmes d'apprentissage automatique peuvent être facilement embarqués sur des appareils connectés, que leurs performances ne font pas défaut et que leur taille et leur consommation d'énergie n'entravent pas le fonctionnement habituel de l'appareil.

La compression des données avec ce type de méthodes de compression avec pertes bornées permet d'étendre la durée de vie de la batterie en fonction de la marge d'erreur acceptée. Nos résultats montrent un excellent taux de compression par rapport à l'état de l'art. Il est à noter que notre schéma évite d'envoyer toute donnée, alors que les mécanismes de compression classiques envoient au moins un en-tête de trame à chaque fois que des données compressées sont envoyées. De plus, ces deux approches sont complémentaires, et les données peuvent être compressées à l'aide des mécanismes de compression classiques une fois que les informations non pertinentes ont été supprimées. Une étude approfondie de la consommation d'énergie a prouvé que notre algorithme économise une partie de l'énergie du dispositif qui peut être utilisée pour d'autres communications. Nous avons développé des arguments concernant les capacités de stockage de poids ML pour les infrastructures IoT soutenues par des approches DNS. Une telle solution semble réalisable en ce qui concerne la taille du fichier de paramètres, mais nécessiterait un travail expérimental pour étayer nos hypothèses par des mesures chiffrées.

Une autre approche consisterait à sélectionner plusieurs caractéristiques sur plusieurs valeurs pour atteindre une plus grande précision dans le calcul avec un recalibrage plus complexe. Des travaux sont menés en contribuant à la communauté TensorFlow Lite actuelle pour proposer un portage complet des bibliothèques LSTM du projet global TensorFlow vers la communauté TensorFlow Lite pour microcontrôleurs. D'autres travaux incluent l'étude de la taille maximale supportée pour les réseaux de neurones afin d'approfondir notre connaissance des réseaux de neurones de petite taille et leurs performances.

En bref, l'utilisation du DNS comme service de base pour l'interconnexion des réseaux, l'hébergement de règles de protocole de communication pour améliorer l'architecture de la solution IoT dans nos différentes expérimentations ont montré des résultats pertinents. Nous avons travaillé à la construction d'une infrastructure d'itinérance, facile à utiliser et fédérée pour interconnecter les réseaux LoRaWAN comme solution pour améliorer l'interopérabilité entre les infrastructures IoT

soutenue par le DNS. Nous avons développé des améliorations au protocole SCHC en hébergeant des règles sur l'infrastructure DNS et en permettant aux éléments réseaux d'interroger une zone DNS globale qui héberge les identifiant des règles et leur numéro de version. Enfin, nous avons développé un algorithme de minimisation de la transmission en intégrant un algorithme d'apprentissage basé sur l'algorithme LSTM sur les capteurs LPWANs et avons étudié son impact sur les données et l'infrastructure sous-jacentes. Les résultats présentés dans cette thèse montrent que le DNS, bien qu'étant l'un des plus anciens protocoles utilisés sur Internet, peut proposer des améliorations pertinentes aux déploiements d'infrastructures et accompagner les nouveaux cas d'usage IoT. Les travaux les plus récents et en cours de la communauté DNS pourraient aider à sécuriser les applications susmentionnées, comme le passage du DNS classique à ses dernières implémentations plus sécurisées, ainsi que confirmer l'authenticité des informations ou aider à la découverte de services pour soutenir les applications IoT.

Bibliography

[1] *How Many Things Are Currently Connected To The "Internet of Things" (IoT)?* https://www.forbes.com/sites/quora/2013/01/07/how-many-things-are-currently-connected-to-the-internet-of-things-iot/. Jan. 2013.

[2] Susan Symington, William Polk, and Murugiah Souppaya. *Trusted Internet of Things (IoT) Device Network-Layer Onboarding and Lifecycle Management (Draft)*. Sept. 2020. DOI: 10.6028/NIST.CSWP.09082020-draft. URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.09082020-draft.pdf.

[3] A. Minaburo et al. "SCHC: Generic Framework for Static Context Header Compression and Fragmentation - RFC 8724". In: *IETF lpwan working group* (Apr. 2020). https://www.rfc-editor.org/info/rfc8724. DOI: 10.17487/RFC8724.

[4] Wael Ayoub et al. "SCHC-Based Solution for Roaming in LoRaWAN". In: *International Conference on Broadband and Wireless Computing, Communication and Applications*. Springer. Nov. 2019, pp. 162–172. DOI: 10.1007/978-3-030-33506-9_15.

[5] S. Cheshire and M. Krochmal. "DNS-Based Service Discovery - RFC 6763". In: *IETF Independant RFC* (Feb. 2013). https://www.rfc-editor.org/info/rfc6763. DOI: 10.17487/RFC6763.

[6] *IoT use-case: The Connected Cow! Yes, really.* https://www.basvankaam.com/2017/04/04/iot-use-case-the-connected-cow-yes-really/. Apr. 2017.

[7] *Computer network naming scheme*. https://en.wikipedia.org/wiki/Computer_network_naming_scheme. 2021.

[8] K. Hubbard et al. "INTERNET REGISTRY IP ALLOCATION GUIDELINES - RFC 2050". In: *IETF Legacy Historic RFC* (Nov. 1996). https://www.rfc-editor.org/info/rfc2050. DOI: 10.17487/RFC2050.

[9] P. Mockapetris. "Domain names - concepts and facilities - STD 13 - RFC 1034". In: *IETF Legacy Internet Standard RFC* (Nov. 1987). https://www.rfc-editor.org/info/rfc1034. DOI: 10.17487/RFC1034.

[10] P. Mockapetris. "Domain names - implementation and specification - STD 13 - RFC 1035". In: *IETF Legacy Internet Standard RFC* (Nov. 1987). https://www.rfc-editor.org/info/rfc1035. DOI: 10.17487/RFC1035.

[11] *DINR - (DNS and Internet Naming Research Directions) Workshop*. https://ant.isi.edu/events/dinr2016/P/p72.pdf. Nov. 2016.

[12] J. Peterson. "Telephone Number Mapping (ENUM) Service Registration for Presence Services - RFC 3953". In: *IETF enum working group* (Jan. 2005). https://www.rfc-editor.org/info/rfc3953. DOI: 10.17487/RFC3953.

[13] *Object Naming Service*. https://www.gs1.org/sites/default/files/docs/epc/ons_2_0_1-standard-20130131.pdf. 2013.

[14] *Approximative number of websites*. https://www.internetlivestats.com/total-number-of-websites/. 2018.

[15] *WBA-IoT-Dynamic-Roaming*. WBA White paper. https://wballiance.com/iot-interoperability-and-roaming-iot-dynamic-roaming/. December 2019.

[16] G. Hatzivasilis et al. "The Interoperability of Things: Interoperable solutions as an enabler for IoT and Web 3.0". In: IEEE 23rd International Workshop on Computer Aided Modeling, Design of Communication Links, and Networks (CAMAD). IEEE, 2018.

[17] "Advancing IoT Platforms Interoperability". In: ed. by Norway Ovidiu Vermesan SINTEF. River Publishers Series in Information Science and Technology, June 2018. DOI: https://doi.org/10.13052/rp-9788770220057.

[18] Ivan Gojmerac et al. "Bridging IoT islands: the symbIoTe project". In: *2nd IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. Springer. September 2016, 315–318. DOI: 10.1007/s00502-016-0439-1.

[19] Debasis Bandyopadhyay and Jaydip Sen. "Internet of Things: Applications and Challenges in Technology and Standardization". In: *Wireless Personal Communications* (May 2011), 49–69. DOI: 10.1007/s11277-011-0288-5.

[20] I. Fosić and K. Šolić. "Graph Database Approach for Data Storing, Presentation and Manipulation". In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2019, pp. 1548–1552. DOI: 10.23919/MIPRO.2019.8756793.

[21] Sharvari Rautmare and D. M. Bhalerao. "MySQL and NoSQL database comparison for IoT application". In: *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*. 2016, pp. 235–238. DOI: 10.1109/ICACA.2016.7887957.

[22] Nazım Yılmaz et al. "Evaluation of storage and query performance of sensor based Internet of Things data with MongoDB". In: *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. 2018, pp. 1–6. DOI: 10.1109/IDAP.2018.8620837.

[23] Yong-Shin Kang et al. "MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data". In: *IEEE Sensors Journal* 16.2 (2016), pp. 485–497. DOI: 10.1109/JSEN.2015.2483499.

[24] Byung Hoo Song et al. "Enhanced query processing using weighted predicate tree in edge computing environment". In: *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. 2017, pp. 48–53. DOI: 10.1109/CSCN.2017.8088597.

[25] Farah Karim et al. "Semantic Enrichment of IoT Stream Data On-demand". In: *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. 2018, pp. 33–40. DOI: 10.1109/ICSC.2018.00014.

[26] Dennis Boldt et al. "SPARQL for Networks of Embedded Systems". In: *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. Vol. 1. 2015, pp. 93–100. DOI: 10.1109/WI-IAT.2015.187.

[27] Farag Azzedin et al. "Systematic Partitioning and Labeling XML Subtrees for Efficient Processing of XML Queries in IoT Environments". In: *IEEE Access* 8 (2020), pp. 61817–61833. DOI: 10.1109/ACCESS.2020.2984600.

[28] Yan Wang et al. "Skyline Preference Query Based on Massive and Incomplete Dataset". In: *IEEE Access* 5 (2017), pp. 3183–3192. DOI: 10.1109/ACCESS.2016.2639558.

[29] Juan A. Colmenares, Reza Dorrigiv, and Daniel G. Waddington. "A single-node datastore for high-velocity multidimensional sensor data". In: *2017*

IEEE International Conference on Big Data (Big Data). 2017, pp. 445–452. DOI: 10.1109/BigData.2017.8257956.

[30] Zhou Zhang et al. "RadixKV: A Memory Efficient and High Performance Key-Value Store". In: *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 2019, pp. 2774–2781. DOI: 10.1109/HPCC/SmartCity/DSS.2019.00389.

[31] Yong Jin et al. "A Lightweight and Secure IoT Remote Monitoring Mechanism Using DNS with Privacy Preservation". In: *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2019, pp. 1–2. DOI: 10.1109/CCNC.2019.8651860.

[32] Kefeng Feng et al. "L-Match: A Lightweight and Effective Subsequence Matching Approach". In: *IEEE Access* 8 (2020), pp. 71572–71583. DOI: 10.1109/ACCESS.2020.2987761.

[33] Jiaye Wu et al. "KV-Match: A Subsequence Matching Approach Supporting Normalization and Time Warping". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 2019, pp. 866–877. DOI: 10.1109/ICDE.2019.00082.

[34] Wenxi Zeng et al. "Monitoring Data Management Services on the Edge Using Enhanced TSDBs". In: *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*. 2019, pp. 9–16. DOI: 10.1109/SOCA.2019.00010.

[35] Xiujun Wang et al. "Near-Optimal Data Structure for Approximate Range Emptiness Problem in Information-Centric Internet of Things". In: *IEEE Access* 7 (2019), pp. 21857–21869. DOI: 10.1109/ACCESS.2019.2897154.

[36] Chuyang Gao, Shuai Zhao, and Bo Cheng. "Design and Implementation of Real Time and History multi-view IoT trend Display and Control System". In: *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. 2019, pp. 1078–1083. DOI: 10.1109/ITAIC.2019.8785795.

[37] Oluwaseun Bamgboye, Xiaodong Liu, and Peter Cruickshank. "Semantic Stream Management Framework for Data Consistency in Smart Spaces". In: *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 2. 2019, pp. 85–90. DOI: 10.1109/COMPSAC.2019.10188.

[38] Shiqiang Li et al. "An Effective Spatio-Temporal Query Framework for Massive Trajectory Data in Urban Computing". In: *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. 2019, pp. 586–593. DOI: 10.1109/ICPADS47876.2019.00089.

[39] Xingsheng Zhao et al. "A Road-Aware Spatial Mapping for Moving Objects". In: *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. 2018, pp. 1–8. DOI: 10.1109/PCCC.2018.8710770.

[40] Santosh Pattar et al. "Location-aware IoT Search Framework based on Data Messaging and Aggregation Techniques". In: *2019 Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE)*. 2019, pp. 138–145. DOI: 10.1109/WITCONECE48374.2019.9092903.

[41] Sungkwang Eom, Sangjin Shin, and Kyong-Ho Lee. "Spatiotemporal query processing for semantic data stream". In: *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*. 2015, pp. 290–297. DOI: 10.1109/ICOSC.2015.7050822.

[42] Karim Fathallah, Mohamed Amine Abid, and Nejib Ben Hadj-Alouane. "Routing Of Spatial Queries Over IOT Enabled Wireless Sensor Networks".

In: *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*. 2019, pp. 1779–1784. DOI: 10.1109/IWCMC.2019.8766512.

[43] Xiaoyan Chen et al. "STaaS: Spatio Temporal Historian as a Service". In: *2015 IEEE International Conference on Web Services*. 2015, pp. 747–750. DOI: 10.1109/ICWS.2015.107.

[44] Charbel El Kaed and Matthieu Boujonnier. "FOrTÉ: A Federated Ontology and Timeseries Query Engine". In: *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2017, pp. 983–990. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.151.

[45] Jine Tang et al. "SMPKR: Search Engine for Internet of Things". In: *IEEE Access* 7 (2019), pp. 163615–163625. DOI: 10.1109/ACCESS.2019.2952390.

[46] Bo Yin and Xuetao Wei. "Communication-Efficient Data Aggregation Tree Construction for Complex Queries in IoT Applications". In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 3352–3363. DOI: 10.1109/JIOT.2018.2882820.

[47] Kostas Kolomvatsos et al. "A Distributed Data Allocation Scheme for Autonomous Nodes". In: *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 2018, pp. 1651–1658. DOI: 10.1109/SmartWorld.2018.00282.

[48] Xiaocui Li and Zhangbing Zhou. "Multi-Attribute Query Processing Through In-Network Aggregation in Edge Computing". In: *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)*. 2018, pp. 144–151. DOI: 10.1109/SKG.2018.00027.

[49] Rajrup Ghosh, Siva Prakash Reddy Komma, and Yogesh Simmhan. "Adaptive Energy-Aware Scheduling of Dynamic Event Analytics Across Edge and Cloud Resources". In: *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. 2018, pp. 72–82. DOI: 10.1109/CCGRID.2018.00022.

[50] Francesco Renna et al. "Query Processing for the Internet-of-Things: Coupling of Device Energy Consumption and Cloud Infrastructure Billing". In: *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 2016, pp. 83–94. DOI: 10.1109/IoTDI.2015.37.

[51] Mert Onuralp Gökalp, Altan Koçyigit, and P. Erhan Eren. "A Cloud Based Architecture for Distributed Real Time Processing of Continuous Queries". In: *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*. 2015, pp. 459–462. DOI: 10.1109/SEAA.2015.61.

[52] Juan Luis Pérez and David Carrera. "Performance Characterization of the Serviotic API: An IoT-as-a-Service Data Management Platform". In: *2015 IEEE First International Conference on Big Data Computing Service and Applications*. 2015, pp. 62–71. DOI: 10.1109/BigDataService.2015.58.

[53] Mayank Tiwary et al. "Introducing Network Multi-Tenancy for Cloud-Based Enterprise Resource Planning: An IoT Application". In: *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*. 2018, pp. 1263–1269. DOI: 10.1109/ISIE.2018.8433724.

[54] Yang Yang, Qiang Cao, and Hong Jiang. "EdgeDB: An Efficient Time-Series Database for Edge Computing". In: *IEEE Access* 7 (2019), pp. 142295–142307. DOI: 10.1109/ACCESS.2019.2943876.

[55] Chuan-Chi Lai et al. "Probabilistic Top- k Dominating Query Monitoring Over Multiple Uncertain IoT Data Streams in Edge Computing Environments". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8563–8576. DOI: 10.1109/JIOT.2019.2920908.

[56] Hessam Moeini et al. "Toward Data Discovery in Dynamic Smart City Applications". In: *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. Aug. 2019, 2572–2579. DOI: 10.1109/HPCC/SmartCity/DSS.2019.00360.

[57] Jonathan Oostvogels et al. "Expressive Feature-oriented Multicast for the Internet of Things". In: *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2019, pp. 173–175. DOI: 10.1109/DCOSS.2019.00046.

[58] Hessam Moeini, I-Ling Yen, and Farokh Bastani. "Service Specification and Discovery in IoT Networks". In: *2019 IEEE International Conference on Web Services (ICWS)*. 2019, pp. 55–59. DOI: 10.1109/ICWS.2019.00021.

[59] Hessam Moeini, I-Ling Yen, and Farokh Bastani. "Efficient Multi-Keyword Based Service Discovery Routing in Peer-to-Peer IoT Networks". In: *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. 2019, pp. 703–710. DOI: 10.1109/ICPADS47876.2019.00104.

[60] Azzam Alsudais et al. "FOCUS: Scalable Search Over Highly Dynamic Geodistributed State". In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 2019, pp. 2134–2144. DOI: 10.1109/ICDCS.2019.00210.

[61] Yugo Nakamura et al. "In-Situ Resource Provisioning with Adaptive Scale-out for Regional IoT Services". In: *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. 2018, pp. 203–213. DOI: 10.1109/SEC.2018.00022.

[62] Po-Yen Chang et al. "Spatial and Temporal Aggregation for Small and Massive Transmissions in LTE-M Networks". In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 2017, pp. 1–5. DOI: 10.1109/WCNC.2017.7925549.

[63] Jine Tang et al. "Using Collaborative Edge-Cloud Cache for Search in Internet of Things". In: *IEEE Internet of Things Journal* 7.2 (2020), pp. 922–936. DOI: 10.1109/JIOT.2019.2946389.

[64] Yang Liu and Yang Zhou. "Development of distributed cache strategy for analytic cluster in an Internet of Things system". In: *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*. 2018, pp. 1–6. DOI: 10.1109/ICNSC.2018.8361300.

[65] Thada Wangthammang and Pichaya Tandayya. "A Software Cache Mechanism for Reducing the OpenTSDB Query Time". In: *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*. 2018, pp. 60–65. DOI: 10.1109/ISCIT.2018.8587857.

[66] Dingding Li et al. "SEER-MCache: A Prefetchable Memory Object Caching System for IoT Real-Time Data Processing". In: *IEEE Internet of Things Journal* 5.5 (2018), pp. 3648–3660. DOI: 10.1109/JIOT.2018.2868334.

[67] Truong Khanh Duy et al. "SemIDEA: Towards a Semantic IoT Data Analytic Framework for Facilitating Environmental Protection". In: *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*. 2019, pp. 481–486. DOI: 10.1109/ISCIT.2019.8905178.

[68] Sondes TITI, Hadda BEN ELHADJ, and Lamia CHAARI. "An ontology-based healthcare monitoring system in the Internet of Things". In: *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*. 2019, pp. 319–324. DOI: 10.1109/IWCMC.2019.8766510.

[69] Maria Ganzha et al. "Streaming semantic translations". In: *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*. 2017, pp. 1–8. DOI: 10.1109/ICSTCC.2017.8107003.

[70] Yongrui Qin, Quan Z. Sheng, and Edward Curry. "Matching Over Linked Data Streams in the Internet of Things". In: *IEEE Internet Computing* 19.3 (2015), pp. 21–27. DOI: 10.1109/MIC.2015.29.

[71] Sara Bonfitto et al. "On the Bulk Ingestion of IoT Devices from Heterogeneous IoT Brokers". In: *2019 IEEE International Congress on Internet of Things (ICIOT)*. July 2019, 189–195. DOI: 10.1109/ICIOT.2019.00039.

[72] Hani Ramadhan et al. "MusQ: A Multi-Store Query System for IoT Data Using a Datalog-Like Language". In: *IEEE Access* 8 (2020), pp. 58032–58056. DOI: 10.1109/ACCESS.2020.2982472.

[73] Shanshan Wu et al. "Storage and retrieval of massive heterogeneous IoT data based on hybrid storage". In: *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 2017, pp. 2982–2987. DOI: 10.1109/FSKD.2017.8393258.

[74] Samir Berrani et al. "A Semantic Model for Service Description in the Internet of Things". In: *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. 2018, pp. 49–54. DOI: 10.1109/SaCoNeT.2018.8585679.

[75] Ruhan Dos Reis et al. "A Soft Real-Time Stream Reasoning Service for the Internet of Things". In: *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. 2019, pp. 166–169. DOI: 10.1109/ICOSC.2019.8665668.

[76] Güngör Yıldırım and Yetkin Tatar. "Simplified Agent-Based Resource Sharing Approach for WSN-WSN Interaction in IoT/CPS Projects". In: *IEEE Access* 6 (2018), pp. 78077–78091. DOI: 10.1109/ACCESS.2018.2884741.

[77] Fadi T. El-Hassan and Dan Ionescu. "Design and Implementation of a Hardware Versatile Publish-Subscribe Architecture for the Internet of Things". In: *IEEE Access* 6 (2018), 31872–31890. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2842706.

[78] Duo Ding, Minbo Li, and Zhu Zhu. "Object Naming Service Supporting Heterogeneous Object Code Identification for IoT System". In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 01. July 2018, 545–554. DOI: 10.1109/COMPSAC.2018.00084.

[79] Willian T. Lunardi et al. "Context-based search engine for industrial IoT: Discovery, search, selection, and usage of devices". In: *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*. 2015, pp. 1–8. DOI: 10.1109/ETFA.2015.7301477.

[80] Dimitrios Georgakopoulos et al. "Towards a RISC Framework for Efficient Contextualisation in the IoT". In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017, pp. 1993–1996. DOI: 10.1109/ICDCS.2017.308.

[81] Alexey Medvedev et al. "Situation Modelling, Representation, and Querying in Context-as-a-Service IoT Platform". In: *2018 Global Internet of Things Summit (GIoTS)*. 2018, pp. 1–6. DOI: 10.1109/GIOTS.2018.8534571.

[82] Agung Prasetio, Sabriansyah Rizqika Akbar, and Bayu Priyambadha. "Implementation of semantic system in the smart home lights device based on

agent". In: *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*. 2017, pp. 93–99. DOI: 10.1109/SIET.2017.8304116.

[83] Sangsu Lee, Tomasz Kalbarczyk, and Christine Julien. "Reminisce: Transparent and Contextually-Relevant Retrospection". In: *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2019, pp. 355–357. DOI: 10.1109/PERCOMW.2019.8730847.

[84] K N Prashanth Kumar, V Ravi Kumar, and K Raghuveer. "A Survey on Semantic Web Technologies for the Internet of Things". In: *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*. 2017, pp. 316–322. DOI: 10.1109/CTCEEC.2017.8454974.

[85] Prem Prakash Jayaraman et al. "Do-it-Yourself Digital Agriculture applications with semantically enhanced IoT platform". In: *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015, pp. 1–6. DOI: 10.1109/ISSNIP.2015.7106951.

[86] Marc-Oliver Pahl and Stefan Liebald. "A Modular Distributed IoT Service Discovery". In: *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 2019, pp. 448–454.

[87] Alireza Hassani et al. "Context-as-a-Service Platform: Exchange and Share Context in an IoT Ecosystem". In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2018, pp. 385–390. DOI: 10.1109/PERCOMW.2018.8480240.

[88] Walid Osamy, Ahmed M. Khedr, and Ahmed Salim. "ADSDA: Adaptive Distributed Service Discovery Algorithm for Internet of Things Based Mobile Wireless Sensor Networks". In: *IEEE Sensors Journal* 19.22 (2019), pp. 10869–10880. DOI: 10.1109/JSEN.2019.2930589.

[89] Mohammed B. M. Kamel, Bruno Crispo, and Peter Ligeti. "A Decentralized and Scalable Model for Resource Discovery in IoT Network". In: *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2019, pp. 1–4. DOI: 10.1109/WiMOB.2019.8923352.

[90] T. Elsaleh et al. "IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams". In: *2019 Global IoT Summit (GIoTS)*. 2019, pp. 1–6. DOI: 10.1109/GIOTS.2019.8766367.

[91] Sejin Chun et al. "Semantic description, discovery and integration for the Internet of Things". In: *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*. 2015, pp. 272–275. DOI: 10.1109/ICOSC.2015.7050819.

[92] Matthew Weber, Ravi Akella, and Edward A. Lee. "Service Discovery for the Connected Car with Semantic Accessors". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 2417–2422. DOI: 10.1109/IVS.2019.8813884.

[93] Steffen Huber et al. "Using Semantic Queries to Enable Dynamic Service Invocation for Processes in the Internet of Things". In: *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. 2016, pp. 214–221. DOI: 10.1109/ICSC.2016.75.

[94] Setiawan Wibowo Purnomo and Fuchun Joseph Lin. "Enhancing Semantic Discovery in oneM2M with Direct Query". In: *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. 2018, pp. 1–6. DOI: 10.1109/CSCN.2018.8581709.

[95] Alexey Medvedev et al. "Benchmarking IoT Context Management Platforms: High-level Queries Matter". In: *2019 Global IoT Summit (GIoTS)*. 2019, pp. 1–6. DOI: 10.1109/GIOTS.2019.8766395.

[96] Maria Bermudez-Edo et al. "IoT-Lite: A Lightweight Semantic Model for the Internet of Things". In: *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. July 2016, 90–97. DOI: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0035.

[97] Charbel El Kaed et al. "SQenloT: Semantic query engine for industrial Internet-of-Things gateways". In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. Dec. 2016, 204–209. DOI: 10.1109/WF-IoT.2016.7845468.

[98] Muhammad Shahzad and Anirudh Ganji. "IoTm: A Lightweight Framework for Fine-Grained Measurements of IoT Performance Metrics". In: *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. 2018, pp. 12–22. DOI: 10.1109/ICNP.2018.00012.

[99] Jie Ding, Rui Wang, and Xiao Chen. "Performance modeling and evaluation of real-time traffic status query for intelligent traffic systems". In: *2016 22nd Asia-Pacific Conference on Communications (APCC)*. 2016, pp. 238–242. DOI: 10.1109/APCC.2016.7581503.

[100] Stefano Rinaldi et al. "Impact of Data Model on Performance of Time Series Database for Internet of Things Applications". In: *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 2019, pp. 1–6. DOI: 10.1109/I2MTC.2019.8827164.

[101] Ji Li et al. "Approximate data aggregation in sensor equipped IoT networks". In: *Tsinghua Science and Technology* 25.1 (2020), pp. 44–55. DOI: 10.26599/TST.2019.9010023.

[102] Lin-Ru Feng, Chuan-Ming Liu, and Chuan-Chi Lai. "Probabilistic reverse nearest neighbors on uncertain data streams". In: *2018 7th International Symposium on Next Generation Electronics (ISNE)*. 2018, pp. 1–4. DOI: 10.1109/ISNE.2018.8394733.

[103] Isam Mashhour Al Jawarneh et al. "Spatial-Aware Approximate Big Data Stream Processing". In: *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9014291.

[104] Mingliu Liu et al. "Sensor Information Retrieval From Internet of Things: Representation and Indexing". In: *IEEE Access* 6 (2018), pp. 36509–36521. DOI: 10.1109/ACCESS.2018.2849865.

[105] Sobhan Jit Muni and Umamani Subudhi. "Selective Harmonic Elimination in Single Phase Inverter using Artificial Neural Network". In: *2018 International Conference on Applied Electromagnetics, Signal Processing and Communication (AESPC)*. Vol. 1. Oct. 2018, 1–6. DOI: 10.1109/AESPC44649.2018.9033365.

[106] Jianwen Ding and Dan Fan. "Edge Computing for Terminal Query Based on IoT". In: *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*. 2019, pp. 70–76. DOI: 10.1109/SmartIoT.2019.00020.

[107] Yongheng Wang and Kening Cao. "Context-aware Complex Event Processing for event cloud in Internet of Things". In: *2012 International Conference on Wireless Communications and Signal Processing (WCSP)*. 2012, pp. 1–6. DOI: 10.1109/WCSP.2012.6542861.

[108] Irfan Mehmood et al. "Efficient Image Recognition and Retrieval on IoT-Assisted Energy-Constrained Platforms From Big Data Repositories". In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 9246–9255. DOI: 10.1109/JIOT.2019.2896151.

[109] Tae-Yeun Kim, Sang-Hyun Bae, and Young-Eun An. "Design of Smart Home Implementation Within IoT Natural Language Interface". In: *IEEE Access* 8 (2020), pp. 84929–84949. DOI: 10.1109/ACCESS.2020.2992512.

[110] Michele Nitti, Virginia Pilloni, and Daniele D. Giusto. "Searching the social Internet of Things by exploiting object similarity". In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. 2016, pp. 371–376. DOI: 10.1109/WF-IoT.2016.7845506.

[111] Younggi Kim and Younghee Lee. "Automatic Generation of Social Relationships between Internet of Things in Smart Home Using SDN-Based Home Cloud". In: *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. 2015, pp. 662–667. DOI: 10.1109/WAINA.2015.93.

[112] Chang-Su LEE et al. "Design and Implementation of Autonomous Collaboration System of Smart Things using accumulated Experience knowledge". In: *2019 21st International Conference on Advanced Communication Technology (ICACT)*. 2019, pp. 305–313. DOI: 10.23919/ICACT.2019.8701947.

[113] M Mazhar Rathore et al. "Exploiting real-time big data to empower smart transportation using big graphs". In: *2016 IEEE Region 10 Symposium (TENSYMP)*. 2016, pp. 135–139. DOI: 10.1109/TENCONSpring.2016.7519392.

[114] Gaurav Tripathi, Bhawna Sharma, and Sonali Rajvanshi. "A combination of Internet of Things (IoT) and graph database for future battlefield systems". In: *2017 International Conference on Computing, Communication and Automation (ICCCA)*. 2017, pp. 1252–1257. DOI: 10.1109/CCAA.2017.8230010.

[115] Yuhai Zhao, Xiangjun Dong, and Ying Yin. "Effective and Efficient Dense Subgraph Query in Large-Scale Social Internet of Things". In: *IEEE Transactions on Industrial Informatics* 16.4 (2020), pp. 2726–2736. DOI: 10.1109/TII.2019.2955144.

[116] Anubha Jain, Trupti Padiya, and Minal Bhise. "Log based method for faster IoT queries". In: *2017 IEEE Region 10 Symposium (TENSYMP)*. 2017, pp. 1–4. DOI: 10.1109/TENCONSpring.2017.8070066.

[117] Aaisha Makkar et al. "QAIR: Quality Assessment Scheme for Information Retrieval in IoT Infrastructures". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647180.

[118] Dimitrios Georgakopoulos. "A Global IoT Device Discovery and Integration Vision". In: *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*. 2019, pp. 214–221. DOI: 10.1109/CIC48465.2019.00035.

[119] Hamza Baqa et al. "Semantic Smart Contracts for Blockchain-based Services in the Internet of Things". In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. 2019, pp. 1–5. DOI: 10.1109/NCA.2019.8935016.

[120] A. Moon et al. "Lossy compression on IoT big data by exploiting spatiotemporal correlation". In: *2017 IEEE High Performance Extreme Computing Conference (HPEC)*. Sept. 2017, pp. 1–7. DOI: 10.1109/HPEC.2017.8091030.

[121] A. Ukil et al. "Adaptive Sensor Data Compression in IoT systems: Sensor data analytics based approach". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2015, pp. 5515–5519. DOI: 10.1109/ICASSP.2015.7179026.

[122] S. Hamdan, A. Awaian, and S. Almajali. "Compression Techniques Used in Iot: A Comparitive Study". In: *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*. Oct. 2019, pp. 1–5. DOI: 10.1109/ICTCS.2019.8923112.

[123] M. Eteläperä, M. Vecchio, and R. Giaffreda. "Improving energy efficiency in IoT with re-configurable virtual objects". In: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. Mar. 2014, pp. 520–525. DOI: 10.1109/WF-IoT.2014.6803222.

[124] M. Hooshmand et al. "Boosting the Battery Life of Wearables for Health Monitoring Through the Compression of Biosignals". In: *IEEE Internet of Things Journal* 4.5 (Oct. 2017), pp. 1647–1662. ISSN: 2372-2541. DOI: 10.1109/JIOT.2017.2689164.

[125] K. Hossain and S. Roy. "A Data Compression and Storage Optimization Framework for IoT Sensor Data in Cloud Storage". In: *2018 21st International Conference of Computer and Information Technology (ICCIT)*. Dec. 2018, pp. 1–6. DOI: 10.1109/ICCITECHN.2018.8631929.

[126] T. Boshita, H. Suzuki, and Y. Matsumoto. "Compression Method of Position Information for IoT-based Bus Location System Using LoRaWAN". In: *2018 Eleventh International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. Oct. 2018, pp. 1–2. DOI: 10.23919/ICMU.2018.8653620.

[127] V. Alieksieiev. "One Approach of Approximation for Incoming Data Stream in IoT Based Monitoring System". In: *2018 IEEE Second International Conference on Data Stream Mining Processing (DSMP)*. Aug. 2018, pp. 94–97. DOI: 10.1109/DSMP.2018.8478466.

[128] E. Guberović et al. "Assessing compression algorithms on IoT sensor nodes". In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. May 2019, pp. 913–918. DOI: 10.23919/MIPRO.2019.8756995.

[129] X. Zhao, V. Sadhu, and D. Pompili. "Analog Signal Compression and Multiplexing Techniques for Healthcare Internet of Things". In: *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. Oct. 2017, pp. 398–406. DOI: 10.1109/MASS.2017.62.

[130] A. Chatterjee, R. J. Shah, and K. S. Hasan. "Efficient Data Compression for IoT Devices using Huffman Coding Based Techniques". In: *2018 IEEE International Conference on Big Data (Big Data)*. Dec. 2018, pp. 5137–5141. DOI: 10.1109/BigData.2018.8622282.

[131] J. Pope et al. "An accelerometer lossless compression algorithm and energy analysis for IoT devices". In: *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. Apr. 2018, pp. 396–401. DOI: 10.1109/WCNCW.2018.8368985.

[132] Y. Xu and T. Kishi. "An Ontology-Based IoT Communication Data Reduction Method". In: *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*. Nov. 2018, pp. 321–325. DOI: 10.1109/UEMCON.2018.8796782.

[133] S. Kartakis et al. "Energy-based adaptive compression in water network control systems". In: *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. Apr. 2016, pp. 43–48. DOI: 10.1109/CySWater.2016.7469055.

[134] Y. Lee, E. Hwang, and J. Choi. "A Unified Approach for Compression and Authentication of Smart Meter Reading in AMI". In: *IEEE Access* 7 (2019), pp. 34383–34394. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2903574.

[135] B. R. Stojkoska and Z. Nikolovski. "Data compression for energy efficient IoT solutions". In: *2017 25th Telecommunication Forum (TELFOR)*. Nov. 2017, pp. 1–4. DOI: 10.1109/TELFOR.2017.8249368.

[136] A. Iikubo and S. Lo. "Design of Power Saving Schemes for the IoT". In: *2019 International Conference on Information Networking (ICOIN)*. Jan. 2019, pp. 316–319. DOI: 10.1109/ICOIN.2019.8718179.

[137] C. J. Deepu, C. Heng, and Y. Lian. "A Hybrid Data Compression Scheme for Power Reduction in Wireless Sensors for IoT". In: *IEEE Transactions on Biomedical Circuits and Systems* 11.2 (Apr. 2017), pp. 245–254. ISSN: 1940-9990. DOI: 10.1109/TBCAS.2016.2591923.

[138] T. L. Le and M. Vo. "Lossless Data Compression Algorithm to Save Energy in Wireless Sensor Network". In: *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*. Nov. 2018, pp. 597–600. DOI: 10.1109/GTSD.2018.8595614.

[139] M. Vecchio, R. Giaffreda, and F. Marcelloni. "Adaptive Lossless Entropy Compressors for Tiny IoT Devices". In: *IEEE Transactions on Wireless Communications* 13.2 (Feb. 2014), pp. 1088–1100. ISSN: 1558-2248. DOI: 10.1109/TWC.2013.121813.130993.

[140] B. Browne, G. Giering, and P. Prestwich. "Pulse-Net: Dynamic Compression of Convolutional Neural Networks". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. Apr. 2019, pp. 346–351. DOI: 10.1109/WF-IoT.2019.8767300.

[141] A. Ukil, S. Bandyopadhyay, and A. Pal. "IoT Data Compression: Sensor-Agnostic Approach". In: *2015 Data Compression Conference*. Apr. 2015, pp. 303–312. DOI: 10.1109/DCC.2015.66.

[142] A. K. M. Al-Qurabat, C. Abou Jaoude, and A. K. Idrees. "Two Tier Data Reduction Technique for Reducing Data Transmission in IoT Sensors". In: *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*. June 2019, pp. 168–173. DOI: 10.1109/IWCMC.2019.8766590.

[143] O. Sarbishei. "Refined Lightweight Temporal Compression for Energy-Efficient Sensor Data Streaming". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. Apr. 2019, pp. 550–553. DOI: 10.1109/WF-IoT.2019.8767351.

[144] J. Park, H. Park, and Y. Choi. "Data compression and prediction using machine learning for industrial IoT". In: *2018 International Conference on Information Networking (ICOIN)*. Jan. 2018, pp. 818–820. DOI: 10.1109/ICOIN.2018.8343232.

[145] K. MATSUDA and M. KUBOTA. "Compound Compression Method for Gathering Traffic of IoT/CPS Data". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. Apr. 2019, pp. 761–766. DOI: 10.1109/WF-IoT.2019.8767326.

[146] C. Hsu, Y. Fang, and F. Yu. "Content-Sensitive Data Compression for IoT Streaming Services". In: *2017 IEEE International Congress on Internet of Things (ICIOT)*. June 2017, pp. 147–150. DOI: 10.1109/IEEE.ICIOT.2017.25.

[147] M. Goyal et al. "DeepZip: Lossless Data Compression Using Recurrent Neural Networks". In: *2019 Data Compression Conference (DCC)*. Mar. 2019, pp. 575–575. DOI: 10.1109/DCC.2019.00087.

[148] Valentina Di Vincenzo, Martin Heusse, and Bernard Tourancheau. "Improving Downlink Scalability in LoRaWAN". In: May 2019, pp. 1–7. DOI: 10.1109/ICC.2019.8761157.

[149] Jessica Lin et al. "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms". In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. DMKD '03. Association for Computing Machinery, 2003, 2–11. ISBN: 9781450374224.

DOI: 10.1145/882082.882086. URL: https://doi.org/10.1145/882082.882086.

[150] T. Schoellhammer et al. "Lightweight temporal compression of microclimate datasets [wireless sensor networks]". In: *29th Annual IEEE International Conference on Local Computer Networks*. Nov. 2004, pp. 516–524. DOI: 10.1109/LCN.2004.72.

[151] S. M. S. Jalaleddine et al. "ECG data compression techniques-a unified approach". In: *IEEE Transactions on Biomedical Engineering* 37.4 (Apr. 1990), pp. 329–343. ISSN: 1558-2531. DOI: 10.1109/10.52340.

[152] E. Zisselman, Amir Adler, and M. Elad. "Compressed Learning for Image Classification: A Deep Neural Network Approach". In: *Handbook of Numerical Analysis*. Vol. 19. Elsevier, Oct. 2018, pp. 3 –17. ISBN: 9780444642059. DOI: 10.1016/bs.hna.2018.08.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S1570865918300024.

[153] E. J. Candès. "Compressive sampling". English. In: *International Congress of Mathematicians, ICM 2006*. Vol. 3. 2006, pp. 1433–1452. URL: www.scopus.com.

[154] E. J. Candes and M. B. Wakin. "An Introduction To Compressive Sampling". In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 21–30. ISSN: 1558-0792. DOI: 10.1109/MSP.2007.914731.

[155] D. L. Donoho. "Compressed sensing". In: *IEEE Transactions on Information Theory* 52.4 (Apr. 2006), pp. 1289–1306. ISSN: 1557-9654. DOI: 10.1109/TIT.2006.871582.

[156] T. Vlašić et al. "Spline-Like Chebyshev Polynomial Representation for Compressed Sensing". In: *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*. Sept. 2019, pp. 135–140. DOI: 10.1109/ISPA.2019.8868926.

[157] D. Del Testa and M. Rossi. "Lightweight Lossy Compression of Biometric Patterns via Denoising Autoencoders". In: *IEEE Signal Processing Letters* 22.12 (Dec. 2015), pp. 2304–2308. ISSN: 1558-2361. DOI: 10.1109/LSP.2015.2476667.

[158] A. Jarwan, A. Sabbah, and M. Ibnkahla. "Data Transmission Reduction Schemes in WSNs for Efficient IoT Systems". In: *IEEE Journal on Selected Areas in Communications* 37.6 (June 2019), pp. 1307–1324. ISSN: 1558-0008. DOI: 10.1109/JSAC.2019.2904357.

[159] T. Xu and I. Darwazeh. "Design and Prototyping of Neural Network Compression for Non-Orthogonal IoT Signals". In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. Apr. 2019, pp. 1–6. DOI: 10.1109/WCNC.2019.8885830.

[160] M. Tomoskozi et al. "Reliable Base Proposal for Header Compression". In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. Sept. 2019, pp. 1–5. DOI: 10.1109/VTCFall.2019.8891332.

[161] *IPv6 over Low Power Wide-Area Networks (lpwan) Working Group*. https://datatracker.ietf.org/wg/lpwan/about/. 2021.

[162] Bart Moons et al. "Using SCHC for an optimized protocol stack in multimodal LPWAN solutions". In: Apr. 2019. DOI: 10.1109/WF-IoT.2019.8767210.

[163] K. Q. Abdelfadeel, V. Cionca, and D. Pesch. "Dynamic Context for Static Context Header compression in LPWANs". In: *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. June 2018, pp. 35–42. DOI: 10.1109/DCOSS.2018.00013.

[164] H. Shah, R. Shrimali, and V. Parikh. "Header Compression and Neighbor Discovery in 6LoWPAN based IoT - a survey". In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Mar. 2016, pp. 306–311. DOI: 10.1109/WiSPNET.2016.7566144.

[165] Sachin Babar et al. "Proposed embedded security framework for Internet of Things (IoT)". In: *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*. Feb. 2011, 1–5. DOI: 10.1109/WIRELESSVITAE.2011.5940923.

[166] O. Garcia-Morchon, S. Kumar, and M. Sethi. "Internet of Things (IoT) Security: State of the Art and Challenges - RFC 8576". In: *IRTF Informational RFC* (Apr. 2019). https://www.rfc-editor.org/info/rfc8576. DOI: 10.17487/RFC8576.

[167] Luigi Coppolino et al. "My Smart Home is Under Attack". In: *2015 IEEE 18th International Conference on Computational Science and Engineering*. Oct. 2015, 145–151. DOI: 10.1109/CSE.2015.28.

[168] Fadele Ayotunde Alaba et al. "Internet of Things security: A survey". In: *Journal of Network and Computer Applications* 88 (June 2017), 10–28. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2017.04.002.

[169] *Sinkhole Attack - an overview | ScienceDirect Topics*. URL: https://www.sciencedirect.com/topics/computer-science/sinkhole-attack.

[170] M. T. Kurniawan and Setiadi Yazid. "Mitigation strategy of sinkhole attack in Wireless Sensor Network". In: *2017 International Workshop on Big Data and Information Security (IWBIS)*. Sept. 2017, 119–125. DOI: 10.1109/IWBIS.2017.8275112.

[171] Emekcan Aras et al. "Exploring the Security Vulnerabilities of LoRa". In: *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*. June 2017, 1–6. DOI: 10.1109/CYBConf.2017.7985777.

[172] Xueying Yang et al. "Security Vulnerabilities in LoRaWAN". In: *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, Apr. 2018, 129–140. ISBN: 978-1-5386-6312-7. DOI: 10.1109/IoTDI.2018.00022. URL: https://ieeexplore.ieee.org/document/8366983/.

[173] Sungmin Hong et al. "SNAIL: an IP-based wireless sensor network approach to the internet of things". In: *IEEE Wireless Communications* 17.6 (Dec. 2010), 34–42. ISSN: 1558-0687. DOI: 10.1109/MWC.2010.5675776.

[174] Simone Cirani, Gianluigi Ferrari, and Luca Veltri. "Enforcing Security Mechanisms in the IP-Based Internet of Things: An Algorithmic Overview". In: *Algorithms* 6.22 (June 2013), 197–226. DOI: 10.3390/a6020197.

[175] Shahid Raza, Thiemo Voigt, and Vilhelm Jutvik. "Lightweight IKEv2: A Key Management Solution for both the Compressed IPsec and the IEEE 802.15.4 Security". In: (), p. 2.

[176] Oscar Garcia-Morchon and Klaus Wehrle. "Modular context-aware access control for medical sensor networks". In: *Proceedings of the 15th ACM symposium on Access control models and technologies*. SACMAT '10. Association for Computing Machinery, June 2010, 129–138. ISBN: 978-1-4503-0049-0. DOI: 10.1145/1809842.1809864. URL: https://doi.org/10.1145/1809842.1809864.

[177] Rodrigo Roman et al. "Key management systems for sensor networks in the context of the Internet of Things". In: *Computers and Electrical*

Engineering. Modern Trends in Applied Security: Architectures, Implementations and Applications 37.2 (Mar. 2011), 147–159. ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2011.01.009.

[178] Anders Fongen. "Identity Management and Integrity Protection in the Internet of Things". In: Sept. 2012. DOI: 10.1109/EST.2012.15.

[179] Xuanxia Yao et al. "A Lightweight Multicast Authentication Mechanism for Small Scale IoT Applications". In: *IEEE Sensors Journal* 13.10 (Oct. 2013), 3693–3701. ISSN: 1558-1748. DOI: 10.1109/JSEN.2013.2266116.

[180] Pedro Moreno Sanchez, Rafa Marin Lopez, and Antonio F. Gomez Skarmeta. "PANATIKI: A Network Access Control Implementation Based on PANA for IoT Devices". In: *Sensors (Basel, Switzerland)* 13.11 (Nov. 2013), 14888–14917. ISSN: 1424-8220. DOI: 10.3390/s131114888.

[181] Thomas Kothmayr et al. "DTLS based security and two-way authentication for the Internet of Things". In: *Ad Hoc Networks* 11.8 (Nov. 2013), 2710–2723. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2013.05.003.

[182] Ankush Singla et al. "Fast and scalable authentication for vehicular internet of things". In: *Proceedings of the 16th Annual Information Security Symposium*. CERIAS '15. CERIAS - Purdue University, Mar. 2015, p. 1.

[183] Simone Cirani et al. "IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios". In: *IEEE Sensors Journal* 15.2 (Feb. 2015), 1224–1234. ISSN: 2379-9153. DOI: 10.1109/JSEN.2014.2361406.

[184] B. Stackpole. "Centralized authentication services (RADIUS, TACACS, DIAMETER)". In: Sixth. Information Security Management Handbook, Jan. 2007, p. 909.

[185] Ali Dehghantanha Kim-Kwang Raymond Choo, ed. *Blockchain Cybersecurity, Trust and Privacy*. Vol. 79. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-38181-3.

[186] "Na Shi et al. "A Blockchain-Empowered AAA Scheme in the Large-Scale HetNet". In: *"Digital Communications and Networks"* (2020). ISSN: 2468-5925.

[187] *Distributed PKI vs Traditional PKI*. https://dzone.com/articles/distributed-pki-vs-traditional-pki. June 2020.

[188] K. Wierenga, S. Winter, and T. Wolniewicz. "The eduroam Architecture for Network Roaming - RFC 7593". In: *IETF Independant RFC* (Sept. 2015). https://www.rfc-editor.org/info/rfc7593. DOI: 10.17487/RFC7593.

[189] D. Forsberg et al. "Protocol for Carrying Authentication for Network Access (PANA) - RFC 5191". In: *IETF pana working group* (May 2008). https://www.rfc-editor.org/info/rfc5191. DOI: 10.17487/RFC5191.

[190] *Protocol for carrying Authentication for Network Access (pana) (Concluded Working Group)*. https://datatracker.ietf.org/wg/pana/about/. 2009.

[191] Stefanie Gerdes, Olaf Bergmann, and Carsten Bormann. *Delegated CoAP Authentication and Authorization Framework (DCAF) - Draft*. https://datatracker.ietf.org/doc/draft-gerdes-ace-dcaf-authorize/04/. Oct. 2015.

[192] Yan Liu and Kun Wang. "Trust control in heterogeneous networks for Internet of Things". In: *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 1. Oct. 2010, pp. V1–632–V1–636. DOI: 10.1109/ICCASM.2010.5620458.

[193] Dong Chen et al. "TRM-IoT: A trust management model based on fuzzy reputation for internet of things". In: *Computer Science and Information Systems* 8.4 (2011), 1207–1228. ISSN: 1820-0214, 2406-1018. DOI: 10.2298/CSIS110303056C.

[194] Michele Nitti et al. "A subjective model for trustworthiness evaluation in the social Internet of Things". In: *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*. Sept. 2012, 18–23. DOI: 10.1109/PIMRC.2012.6362662.

[195] Zhou Quan et al. "Trusted architecture for farmland wireless sensor networks". In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. Dec. 2012, 782–787. DOI: 10.1109/CloudCom.2012.6427496.

[196] Fenye Bao and Ing-Ray Chen. "Trust management for the internet of things and its application to service composition". In: *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. June 2012, 1–6. DOI: 10.1109/WoWMoM.2012.6263792.

[197] Fenye Bao, Ing-Ray Chen, and Jia Guo. "Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems". In: *2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*. Mar. 2013, 1–7. DOI: 10.1109/ISADS.2013.6513398.

[198] Dennis Gessner et al. "Trustworthy Infrastructure Services for a Secure and Privacy-Respecting Internet of Things". In: *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. June 2012, 998–1003. DOI: 10.1109/TrustCom.2012.286.

[199] Zheng Yan, Peng Zhang, and Athanasios V. Vasilakos. "A survey on trust management for Internet of Things". In: *Journal of Network and Computer Applications* 42 (June 2014), 120–134. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2014.01.014.

[200] Tigist Abera et al. "Invited - Things, trouble, trust: on building trust in IoT systems". In: *Proceedings of the 53rd Annual Design Automation Conference*. DAC '16. Association for Computing Machinery, June 2016, 1–6. ISBN: 978-1-4503-4236-0. DOI: 10.1145/2897937.2905020. URL: https://doi.org/10.1145/2897937.2905020.

[201] Yi-Ting Chiang et al. "Secrecy of Two-Party Secure Computation". In: *Data and Applications Security XIX*. Ed. by Sushil Jajodia and Duminda Wijesekera. Lecture Notes in Computer Science. Springer, 2005, 114–123. ISBN: 978-3-540-31937-5. DOI: 10.1007/11535706_9.

[202] Georgios V. Lioudakis et al. "A Proxy for Privacy: the Discreet Box". In: *EUROCON 2007 - The International Conference on "Computer as a Tool"*. Sept. 2007, 966–973. DOI: 10.1109/EURCON.2007.4400521.

[203] Diego M. Mendez, Ioannis Papapanagiotou, and Baijian Yang. "Internet of Things: Survey on Security and Privacy". In: *Information Security Journal: A Global Perspective* 27.3 (May 2018). arXiv: 1707.01879, 162–182. ISSN: 1939-3555, 1939-3547. DOI: 10.1080/19393555.2018.1458258.

[204] Xin Huang et al. "User interactive Internet of things privacy preserved access control". In: *2012 International Conference for Internet Technology and Secured Transactions*. Dec. 2012, 597–602.

[205] Scott R. Peppet. *Regulating the Internet of Things: First Steps Toward Managing Discrimination, Privacy, Security and Consent*. ID 2409074. Mar. 2014. URL: https://papers.ssrn.com/abstract=2409074.

[206] Sylvain Kubler, Kary Främling, and Andrea Buda. "A standardized approach to deal with firewall and mobility policies in the IoT". In: *Pervasive and Mobile Computing* 20 (July 2015), 100–114. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2014.09.005.

[207] Chengzhe Lai et al. "CPAL: A Conditional Privacy-Preserving Authentication With Access Linkability for Roaming Service". In: *IEEE Internet of Things Journal* 1.1 (Feb. 2014), 46–57. ISSN: 2372-2541. DOI: 10.1109/JIOT.2014.2306673.

[208] Zhiping Wan et al. "An Internet of Things Roaming Authentication Protocol Based on Heterogeneous Fusion Mechanism". In: *IEEE Access* 8 (2020), 17663–17672. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2967469.

[209] Sergey Andreev et al. "Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap". In: *IEEE Communications Magazine* 53.9 (Sept. 2015), pp. 32–40. ISSN: 1558-1896. DOI: 10.1109/MCOM.2015.7263370.

[210] Ramon Sanchez-Iborra and Maria-Dolores Cano. "State of the Art in LP-WAN Solutions for Industrial IoT Services". en. In: *Sensors* 16.5 (May 2016), p. 708. DOI: 10.3390/s16050708. URL: https://www.mdpi.com/1424-8220/16/5/708 (visited on 02/18/2020).

[211] Jean-Paul Bardyn et al. "IoT: The era of LPWAN is starting now". en. In: *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. Lausanne, Switzerland: IEEE, Sept. 2016, pp. 25–30. ISBN: 978-1-5090-2972-3. DOI: 10.1109/ESSCIRC.2016.7598235. URL: http://ieeexplore.ieee.org/document/7598235/ (visited on 12/02/2019).

[212] Samuela Persia, Claudia Carciofi, and Manuel Faccioli. "NB-IoT and LoRA connectivity analysis for M2M/IoT smart grids applications". In: *2017 AEIT International Annual Conference*. ISSN: null. Sept. 2017, pp. 1–6. DOI: 10.23919/AEIT.2017.8240558.

[213] Reza Tadayoni, Anders Henten, and Morten Falch. "Internet of Things — The battle of standards". en. In: *2017 Internet of Things Business Models, Users, and Networks*. Copenhagen: IEEE, Nov. 2017, pp. 1–7. ISBN: 978-1-5386-3197-3. DOI: 10.1109/CTTE.2017.8260927. URL: https://ieeexplore.ieee.org/document/8260927/ (visited on 12/04/2019).

[214] Maria Rita Palattella and Nicola Accettura. "Enabling Internet of Everything Everywhere: LPWAN with Satellite Backhaul". en. In: *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. Thessaloniki, Greece: IEEE, Oct. 2018, pp. 1–5. ISBN: 978-1-5386-7272-3. DOI: 10.1109/GIIS.2018.8635663. URL: https://ieeexplore.ieee.org/document/8635663/ (visited on 12/04/2019).

[215] Wael Ayoub et al. "Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility". In: *IEEE Communications Surveys Tutorials* 21.2 (2019), pp. 1561–1581. ISSN: 2373-745X. DOI: 10.1109/COMST.2018.2877382.

[216] Takwa Attia et al. "Experimental Characterization of LoRaWAN Link Quality". In: Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013371.

[217] Michele Zorzi et al. "From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view". In: *IEEE Wireless Communications* 17.6 (Dec. 2010), pp. 44–51. ISSN: 1558-0687. DOI: 10.1109/MWC.2010.5675777.

[218] Somayya Madakam $IT Applications Group et al. "Internet of Things (IoT): A Literature Review". en. In: *Journal of Computer and Communications* 03.05 (2015), p. 164. DOI: 10.4236/jcc.2015.35021. URL: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=56616\&\#abstract (visited on 02/18/2020).

[219] Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. "Long-Range IoT Technologies: The Dawn of LoRa™". en. In: *Future Access Enablers for Ubiquitous and Intelligent Infrastructures*. Ed. by Vladimir Atanasovski and Alberto Leon-Garcia. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2015, pp. 51–58. ISBN: 978-3-319-27072-2. DOI: 10.1007/978-3-319-27072-2_7.

[220] Lorenzo Vangelista and Marco Centenaro. "Worldwide Connectivity for the Internet of Things Through LoRaWAN". en. In: *Future Internet* 11.3 (Mar. 2019), p. 57. ISSN: 1999-5903. DOI: 10.3390/fi11030057. URL: https://www.mdpi.com/1999-5903/11/3/57 (visited on 12/04/2019).

[221] Vartika Agarwal, Sachin Sharma, and Piyush Agarwal. "IoT Based Smart Transport Management and Vehicle-to-Vehicle Communication System". In: *Computer Networks, Big Data and IoT*. Singapore: Springer Singapore, 2021, pp. 709–716. ISBN: 978-981-16-0965-7.

[222] Andrzej Duda and Martin Heusse. "Spatial Issues in Modeling LoRaWAN Capacity". In: *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWIM '19. Miami Beach, FL, USA: Association for Computing Machinery, Nov. 2019, pp. 191–198. ISBN: 978-1-4503-6904-6. DOI: 10.1145/3345768.3355932. URL: https://doi.org/10.1145/3345768.3355932 (visited on 03/11/2020).

[223] Safwan M. Ghaleb et al. "Mobility management for IoT: a survey". In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (July 2016), p. 165. ISSN: 1687-1499. DOI: 10.1186/s13638-016-0659-4. URL: https://doi.org/10.1186/s13638-016-0659-4 (visited on 02/18/2020).

[224] Sameh Ben Fredj et al. "A Scalable IoT Service Search Based on Clustering and Aggregation". In: *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. ISSN: null. Aug. 2013, pp. 403–410. DOI: 10.1109/GreenCom-iThings-CPSCom.2013.86.

[225] Sufyan Almajali et al. "A framework for efficient and secured mobility of IoT devices in mobile edge computing". In: *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*. ISSN: null. Apr. 2018, pp. 58–62. DOI: 10.1109/FMEC.2018.8364045.

[226] Akram Hakiri et al. "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications". In: *IEEE Communications Magazine* 53.9 (Sept. 2015), pp. 48–54. ISSN: 1558-1896. DOI: 10.1109/MCOM.2015.7263372.

[227] Jorge E. Luzuriaga et al. "Handling mobility in IoT applications using the MQTT protocol". In: *2015 Internet Technologies and Applications (ITA)*. ISSN: null. Sept. 2015, pp. 245–250. DOI: 10.1109/ITechA.2015.7317403.

[228] Di Wu et al. "UbiFlow: Mobility management in urban-scale software defined IoT". In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. ISSN: 0743-166X. Apr. 2015, pp. 208–216. DOI: 10.1109/INFOCOM.2015.7218384.

[229] Samaresh Bera, Sudip Misra, and Mohammad S. Obaidat. "Mobility-Aware Flow-Table Implementation in Software-Defined IoT". In: *2016 IEEE Global Communications Conference (GLOBECOM)*. ISSN: null. Dec. 2016, pp. 1–6. DOI: 10.1109/GLOCOM.2016.7841995.

[230] Maroua Meddeb et al. "AFIRM: Adaptive forwarding based link recovery for mobility support in NDN/IoT networks". en. In: *Future Generation Computer*

Systems 87 (Oct. 2018), pp. 351–363. ISSN: 0167-739X. DOI: 10.1016/j.future.2018.04.087. URL: http://www.sciencedirect.com/science/article/pii/S0167739X17314528 (visited on 02/18/2020).

[231] Arnaud Durand, Pascal Gremaud, and Jacques Pasquier. "Resilient, crowd-sourced LPWAN infrastructure using blockchain". en. In: *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems - CryBlock'18*. Munich, Germany: ACM Press, 2018, pp. 25–29. ISBN: 978-1-4503-5838-5. DOI: 10.1145/3211933.3211938. URL: http://dl.acm.org/citation.cfm?doid=3211933.3211938 (visited on 11/28/2019).

[232] Mehdi Bezahaf, Gaëtan Cathelain, and Tony Ducrocq. "BcWAN: A Federated Low-Power WAN for the Internet of Things (Industry track)". In: Dec. 2018, pp. 54–60. DOI: 10.1145/3284028.3284036.

[233] Seung-Man Chun and Jong-Tae Park. "Mobile CoAP for IoT mobility management". In: *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. ISSN: 2331-9860. Jan. 2015, pp. 283–289. DOI: 10.1109/CCNC.2015.7157990.

[234] Jaeyeon Jung et al. "DNS Performance and the Effectiveness of Caching". In: (Nov. 2001), p. 14.

[235] Peter B. Danzig, Katia Obraczka, and Anant Kumar. "An analysis of wide-area name server traffic: a study of the Internet Domain Name System". In: *ACM SIGCOMM Computer Communication Review* 22.4 (Oct. 1992), 281–292. ISSN: 0146-4833. DOI: 10.1145/144191.144301.

[236] Eric Osterweil et al. "Behavior of DNS' Top Talkers, a .com/.net View". In: *Passive and Active Measurement*. Ed. by Nina Taft and Fabio Ricciato. Vol. 7192. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, 211–220. ISBN: 978-3-642-28536-3. DOI: 10.1007/978-3-642-28537-0_21. URL: http://link.springer.com/10.1007/978-3-642-28537-0_21.

[237] Vasileios Pappas et al. "Impact of Configuration Errors on DNS Robustness". In: (May 2009), p. 12.

[238] Su, Z. and J. Postel. "The Domain Naming Convention for Internet User Applications - RFC 819". In: *IETF Legacy RFC* (Aug. 1982). https://www.rfc-editor.org/info/rfc819. DOI: 10.17487/RFC0819.

[239] Partridge, C. "Mail routing and the domain system - STD 10 - RFC 974". In: *IETF Legacy Historic RFC* (Jan. 1986). https://www.rfc-editor.org/info/rfc974. DOI: 10.17487/RFC0974.

[240] C. Everhart et al. "New DNS RR Definitions - RFC 1183". In: *IETF Legacy Experimental RFC* (Oct. 1990). https://www.rfc-editor.org/info/rfc1183. DOI: 10.17487/RFC1183.

[241] A. Gulbrandsen and P. Vixie. "A DNS RR for specifying the location of services (DNS SRV) - RFC 2052". In: *IETF Legacy Experimental RFC* (Oct. 1996). https://www.rfc-editor.org/info/rfc2052. DOI: 10.17487/RFC2052.

[242] R. Daniel and M. Mealling. "Resolution of Uniform Resource Identifiers using the Domain Name System - RFC 2168". In: *IETF urn working group* (June 1997). https://www.rfc-editor.org/info/rfc2168. DOI: 10.17487/RFC2168.

[243] R. Elz and R. Bush. "Clarifications to the DNS Specification - RFC 2181". In: *IETF dnsind working group* (July 1997). https://www.rfc-editor.org/info/rfc2181. DOI: 10.17487/RFC2181.

[244] P. Beertema. "Common DNS Data File Configuration Errors - RFC 1537". In: *IETF dns working group* (Oct. 1993). https://www.rfc-editor.org/info/rfc1537. DOI: 10.17487/RFC1537.

[245] D. Barr. "Common DNS Operational and Configuration Errors - RFC 1912". In: *IETF Legacy Informational RFC* (Feb. 1996). https://www.rfc-editor.org/info/rfc1912. DOI: 10.17487/RFC1912.

[246] B. Manning and P. Vixie. "Operational Criteria for Root Name Servers - RFC 2010". In: *IETF Legacy Informational RFC* (Oct. 1996). https://www.rfc-editor.org/info/rfc2010. DOI: 10.17487/RFC2010.

[247] M. Hamilton and R. Wright. "Use of DNS Aliases for Network Services - BCP 17 - RFC 2219". In: *IETF ids working group* (Oct. 1997). https://www.rfc-editor.org/info/rfc2219. DOI: 10.17487/RFC2219.

[248] Bert Hubert. *DNS Camel Viewer*. https://powerdns.org/dns-camel/. Aug. 2021.

[249] Bert Hubert. *Herding the DNS Camel*. https://www.ietf.org/blog/herding-dns-camel/. Nov. 2018.

[250] Z. Hu et al. "Specification for DNS over Transport Layer Security (TLS) - RFC 7858". In: *IETF dprive working group* (May 2016). https://www.rfc-editor.org/info/rfc7858. DOI: 10.17487/RFC7858.

[251] T. Reddy, D. Wing, and P. Patil. "DNS over Datagram Transport Layer Security (DTLS) - RFC 8094". In: *IETF dprive working group* (Feb. 2017). https://www.rfc-editor.org/info/rfc8094. DOI: 10.17487/RFC8094.

[252] S. Bortzmeyer. "DNS Privacy Considerations - RFC 7626". In: *IETF dprive working group* (Aug. 2015). https://www.rfc-editor.org/info/rfc7626. DOI: 10.17487/RFC7626.

[253] P. Hoffman and P. McManus. "DNS Queries over HTTPS (DoH) - RFC 8484". In: *IETF doh working group* (Oct. 2018). https://www.rfc-editor.org/info/rfc8484. DOI: 10.17487/RFC8484.

[254] Christian Huitema, Sara Dickinson, and Allison Mankin. "Specification of DNS over Dedicated QUIC Connections - Draft". In: *IETF dprive working group* (July 2021). https://datatracker.ietf.org/doc/draft-ietf-dprive-dnsoquic/03/.

[255] Austin Hounsel et al. "Comparing the Effects of DNS, DoT, and DoH on Web Performance". In: *Proceedings of The Web Conference 2020*. WWW '20. Association for Computing Machinery, Apr. 2020, 562–572. ISBN: 978-1-4503-7023-3. DOI: 10.1145/3366423.3380139. URL: https://doi.org/10.1145/3366423.3380139.

[256] Trinh Viet Doan, Irina Tsareva, and Vaibhav Bajpai. "Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times". In: *Passive and Active Measurement*. Ed. by Oliver Hohlfeld, Andra Lutu, and Dave Levin. Vol. 12671. Lecture Notes in Computer Science. Springer International Publishing, 2021, 192–209. ISBN: 978-3-030-72581-5. DOI: 10.1007/978-3-030-72582-2_12. URL: https://link.springer.com/10.1007/978-3-030-72582-2_12.

[257] R. Arends et al. "DNS Security Introduction and Requirements - RFC 4033". In: *IETF dnsext working group* (Mar. 2005). https://www.rfc-editor.org/info/rfc4033. DOI: 10.17487/RFC4033.

[258] R. Arends et al. "Resource Records for the DNS Security Extensions - RFC 4034". In: *IETF dnsext working group* (Mar. 2005). https://www.rfc-editor.org/info/rfc4034. DOI: 10.17487/RFC4034.

[259] R. Arends et al. "Protocol Modifications for the DNS Security Extensions - RFC 4035". In: *IETF dnsext working group* (Mar. 2005). https://www.rfc-editor.org/info/rfc4035. DOI: 10.17487/RFC4035.

[260] S. Josefsson. "Storing Certificates in the Domain Name System (DNS) - RFC 4398". In: *IETF dnsext working group* (Mar. 2006). https://www.rfc-editor.org/info/rfc4398. DOI: 10.17487/RFC4398.

[261] B. Laurie et al. "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence - RFC 5155". In: *IETF dnsext working group* (Mar. 2008). https://www.rfc-editor.org/info/rfc5155. DOI: 10.17487/RFC5155.

[262] P. Hoffman. "Cryptographic Algorithm Identifier Allocation for DNSSEC - RFC 6014". In: *IETF dnsext working group* (Nov. 2010). https://www.rfc-editor.org/info/rfc6014. DOI: 10.17487/RFC6014.

[263] R. Barnes. "Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE) - RFC 6394". In: *IETF dane working group* (Oct. 2011). https://www.rfc-editor.org/info/rfc6394. DOI: 10.17487/RFC6394.

[264] P. Hoffman and J. Schlyter. "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA - RFC 6698". In: *IETF dane working group* (Aug. 2012). https://www.rfc-editor.org/info/rfc6698. DOI: 10.17487/RFC6698.

[265] O. Gudmundsson. "Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE) - RFC 7218". In: *IETF dane working group* (Apr. 2014). https://www.rfc-editor.org/info/rfc7218. DOI: 10.17487/RFC7218.

[266] V. Dukhovni and W. Hardaker. "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance - RFC 7671". In: *IETF dane working group* (Oct. 2015). https://www.rfc-editor.org/info/rfc7671. DOI: 10.17487/RFC7671.

[267] V. Dukhovni and W. Hardaker. "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) - RFC 7672". In: *IETF dane working group* (Oct. 2015). https://www.rfc-editor.org/info/rfc7672. DOI: 10.17487/RFC7672.

[268] T. Finch, M. Miller, and P. Saint-Andre. "Using DNS-Based Authentication of Named Entities (DANE) TLSA Records with SRV Records - RFC 7673". In: *IETF dane working group* (Oct. 2015). https://www.rfc-editor.org/info/rfc7673. DOI: 10.17487/RFC7673.

[269] P. Wouters. "DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP - RFC 7929". In: *IETF dane working group* (Aug. 2016). https://www.rfc-editor.org/info/rfc7929. DOI: 10.17487/RFC7929.

[270] *Extensions for Scalable DNS Service Discovery (dnssd) Working Group*. https://datatracker.ietf.org/wg/dnssd/about/. 2021.

[271] *Zero Configuration Networking (zeroconf) (Concluded Working Group)*. https://datatracker.ietf.org/wg/zeroconf/about/. 2005.

[272] T. Pusateri and S. Cheshire. "DNS Push Notifications - RFC 8765". In: *IETF dnssd working group* (June 2020). https://www.rfc-editor.org/info/rfc8765. DOI: 10.17487/RFC8765.

[273] S. Cheshire. "Discovery Proxy for Multicast DNS-Based Service Discovery - RFC 8766". In: *IETF dnssd working group* (June 2020). https://www.rfc-editor.org/info/rfc8766. DOI: 10.17487/RFC8766.

[274] K. Lynn et al. "Requirements for Scalable DNS-Based Service Discovery (DNS-SD) / Multicast DNS (mDNS) Extensions - RFC 7558". In: *IETF dnssd*

working group (July 2015). https://www.rfc-editor.org/info/rfc7558. DOI: 10.17487/RFC7558.

[275] Sullivan, A. "Selecting Labels for Use with Conventional DNS and Other Resolution Systems in DNS-Based Service Discovery - RFC 8222". In: *IETF dnssd working group* (Sept. 2017). https://www.rfc-editor.org/info/rfc8222. DOI: 10.17487/RFC8222.

[276] C. Huitema and D. Kaiser. "DNS-Based Service Discovery (DNS-SD) Privacy and Security Requirements - RFC 8882". In: *IETF dnssd working group* (Sept. 2020). https://www.rfc-editor.org/info/rfc8882. DOI: 10.17487/RFC8882.

[277] S. Cheshire, B. Aboba, and E. Guttman. "Dynamic Configuration of IPv4 Link-Local Addresses - RFC 3927". In: *IETF zeroconf working group* (May 2005). https://www.rfc-editor.org/info/rfc3927. DOI: 10.17487/RFC3927.

[278] Antonio J. Jara, Pedro Martinez-Julia, and Antonio Skarmeta. "Light-Weight Multicast DNS and DNS-SD (lmDNS-SD): IPv6-Based Resource and Service Discovery for the Web of Things". In: *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, July 2012, 731–738. ISBN: 978-1-4673-1328-5. DOI: 10.1109/IMIS.2012.200. URL: http://ieeexplore.ieee.org/document/6296945/.

[279] Badis Djamaa and Mark Richardson. "Towards Scalable DNS-Based Service Discovery for the Internet of Things". In: *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*. Ed. by Ramón Hervás et al. Lecture Notes in Computer Science. Springer International Publishing, 2014, 432–435. ISBN: 978-3-319-13102-3. DOI: 10.1007/978-3-319-13102-3_70.

[280] Milosh Stolikj et al. "Proxy support for service discovery using mDNS/DNS-SD in low power networks". In: *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. June 2014, 1–6. DOI: 10.1109/WoWMoM.2014.6918925.

[281] Milosh Stolikj et al. "Context based service discovery in unmanaged networks using mDNS/DNS-SD". In: *2016 IEEE International Conference on Consumer Electronics (ICCE)*. Jan. 2016, 163–165. DOI: 10.1109/ICCE.2016.7430565.

[282] Ahmed Ismail and Wolfgang Kastner. "Discovery in SOA-governed industrial middleware with mDNS and DNS-SD". In: *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Sept. 2016, 1–8. DOI: 10.1109/ETFA.2016.7733529.

[283] Daniel Kaiser et al. "User-Friendly, Versatile, and Efficient Multi-link DNS Service Discovery". In: *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. June 2016, 146–155. DOI: 10.1109/ICDCSW.2016.34.

[284] Seokhwa Kim, Keuntae Lee, and Jaehoon Paul Jeong. "DNS naming services for service discovery and remote control for Internet-of-Things devices". In: *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, Oct. 2017, 1156–1161. ISBN: 978-1-5090-4032-2. DOI: 10.1109/ICTC.2017.8190884. URL: http://ieeexplore.ieee.org/document/8190884/.

[285] Ronny Klauck. "Seamless Integration of Smart Objects into the Internet Using XMPP and mDNS/DNS-SD". In: (), p. 152.

[286] Wen Sun, Jiajia Liu, and Yanlin Yue. "AI-Enhanced Offloading in Edge Computing: When Machine Learning Meets Industrial IoT". In: *IEEE Network* 33.5 (Sept. 2019), 68–74. ISSN: 1558-156X. DOI: 10.1109/MNET.001.1800510.

[287] Seraphin B. Calo et al. "Edge computing architecture for applying AI to IoT". In: *2017 IEEE International Conference on Big Data (Big Data)*. Dec. 2017, 3012–3016. DOI: 10.1109/BigData.2017.8258272.

[288] He Li, Kaoru Ota, and Mianxiong Dong. "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing". In: *IEEE Network* 32.1 (Jan. 2018), 96–101. ISSN: 1558-156X. DOI: 10.1109/MNET.2018.1700202.

[289] Zhihan Lv et al. "AI-enabled IoT-Edge Data Analytics for Connected Living". In: *ACM Transactions on Internet Technology* 21.4 (July 2021), 104:1–104:20. ISSN: 1533-5399. DOI: 10.1145/3421510.

[290] Purnendu Shekhar Pandey. "Machine Learning and IoT for prediction and detection of stress". In: *2017 17th International Conference on Computational Science and Its Applications (ICCSA)*. July 2017, 1–5. DOI: 10.1109/ICCSA.2017.8000018.

[291] Dr. Sivaganesan D. "DESIGN AND DEVELOPMENT AI-ENABLED EDGE COMPUTING FOR INTELLIGENT-IOT APPLICATIONS". In: *Journal of Trends in Computer Science and Smart Technology* 2019.02 (Dec. 2019), 84–94. ISSN: 2582-4104. DOI: 10.36548/jtcsst.2019.2.002.

[292] Olivier Debauche et al. "A new Edge Architecture for AI-IoT services deployment". In: *Procedia Computer Science*. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology 175 (Jan. 2020), 10–19. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.07.006.

[293] Olivier Debauche et al. "Edge AI-IoT Pivot Irrigation, Plant Diseases, and Pests Identification". In: *Procedia Computer Science*. The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) / The 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) / Affiliated Workshops 177 (Jan. 2020), 40–48. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.10.009.

[294] Francesco Piccialli et al. "A machine learning approach for IoT cultural data". In: *Journal of Ambient Intelligence and Humanized Computing* (Sept. 2019). ISSN: 1868-5145. DOI: 10.1007/s12652-019-01452-6. URL: https://doi.org/10.1007/s12652-019-01452-6.

[295] Erwin Adi et al. "Machine learning and data analytics for the IoT". In: *Neural Computing and Applications* 32.20 (Oct. 2020), 16205–16233. ISSN: 1433-3058. DOI: 10.1007/s00521-020-04874-y.

[296] Janice Cañedo and Anthony Skjellum. "Using machine learning to secure IoT systems". In: *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. Dec. 2016, 219–222. DOI: 10.1109/PST.2016.7906930.

[297] Liang Xiao et al. "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?" In: *IEEE Signal Processing Magazine* 35.5 (Sept. 2018), 41–49. ISSN: 1558-0792. DOI: 10.1109/MSP.2018.2825478.

[298] Ibrahim Alrashdi et al. "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. Jan. 2019, 0305–0310. DOI: 10.1109/CCWC.2019.8666450.

[299] Zhihan Lv et al. "AI-empowered IoT Security for Smart Cities". In: *ACM Transactions on Internet Technology* 21.4 (July 2021), 99:1–99:21. ISSN: 1533-5399. DOI: 10.1145/3406115.

[300] Benjamin Sliwa, Nico Piatkowski, and Christian Wietfeld. "LIMITS: Lightweight Machine Learning for IoT Systems with Resource Limitations". In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. June 2020, 1–7. DOI: 10.1109/ICC40277.2020.9149180.

[301] M. Sethi, B. Sarikaya, and D. Garcia-Carrillo. "Secure IoT Bootstrapping: A Survey". In: *IETF T2TRG Draft* (2021). URL: https://tools.ietf.org/html/draft-sarikaya-t2trg-sbootstrapping-11.

[302] *IoT Onboarding Mail Archive discussion*. https://mailarchive.ietf.org/arch/browse/iot-onboarding/?gbt=1&index=XHSahBLvpUU8Sasvt77QLX43OhQ. December 2019.

[303] *LoRaWAN Backend Specifications*. https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf.

[304] Farrell, S., Ed. "Low-Power Wide Area Network (LPWAN) Overview - RFC 8376". In: *IETF lpwan working group* (May 2018). https://www.rfc-editor.org/info/rfc8376. DOI: 10.17487/RFC8376.

[305] Jansen Liando et al. "Known and UnknownFacts of LoRa: Experiences from a Large-scale Measurement Study". In: *ACM Transactions on SensorNetworks* (2019). DOI: 10.1145/3293534.

[306] *Eduroam Website*. https://www.eduroam.org/.

[307] D. Eastlake 3rd. "Domain Name System Security Extensions - RFC 2535". In: *IETF dnssec working group* (Mar. 1999). https://www.rfc-editor.org/info/rfc2535. DOI: 10.17487/RFC2535.

[308] E. M. Torroglosa-Garcia et al. "Enabling Roaming Across Heterogeneous IoT Wireless Networks: LoRaWAN MEETS 5G". In: *IEEE Access* 8 (2020), pp. 103164–103180.

[309] *OpenRoaming, Wireless Broadband Alliance*. https://wballiance.com/openroaming/. 2020.

[310] *"IoT New Vertical Value Chains and Interoperability", Wireless Broadband Alliance (WBA)*. https://www.wballiance.com/wp-content/uploads/2017/03/IoT-New-Vertical-Value-Chains-and-Interoperability-v1.00.pdf. 2020.

[311] Alliance for Internet of Things Innovation (AIOTI). *Identifiers in Internet of Things*. https://aioti.eu/wp-content/uploads/2018/03/AIOTI-Identifiers_in_IoT-1_0.pdf.pdf. Feb. 2018.

[312] Haris Aftab et al. "Analysis of identifiers in IoT platforms". In: *Digital Communications and Networks* 6.3 (2020), pp. 333–340. ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2019.05.003. URL: https://www.sciencedirect.com/science/article/pii/S2352864818300671.

[313] GS1. *EPC Tag Data Standard*. https://www.gs1.org/sites/default/files/docs/epc/GS1_EPC_TDS_i1_11.pdf.

[314] GSMA. *EPS Roaming Guidelines Version 22.0*. 2020. URL: \url{https://www.gsma.com/newsroom/wp-content/uploads/IR.88-v22.0-2.pdf}.

[315] *IoTRoam Proof of Concept*. https://github.com/afnic/IoTRoam-Tutorial/. 2020.

[316] *IoTRoam QuickStart Tutorial*. https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/QuickStart.md.

[317] *Video Afnic*. https://iot.rd.nic.fr/Video/version3.mp4. 2020.

[318] *Chirpstack Website*. https://chirpstack.io. 2021.

[319] Kazunori Fujiwara, Akira Sato, and Kenichi Yoshida. "DNS Traffic Analysis — CDN and the World IPv6 Launch". In: *Journal of Information Processing* 21.3 (2013), 517–526. DOI: 10.2197/ipsjjip.21.517.

[320] Mark Allman. "Putting DNS in Context". In: *Proceedings of the ACM Internet Measurement Conference*. IMC '20. Association for Computing Machinery, Oct. 2020, 309–316. ISBN: 978-1-4503-8138-3. DOI: 10.1145/3419394.3423659. URL: https://doi.org/10.1145/3419394.3423659.

[321] *DNS Prefetching - The Chromium Projects*. http://www.chromium.org/developers/design-documents/dns-prefetching..

[322] Acklio. *IPv6 over IoT networks*. https://www.ackl.io/technology/schc/. Aug. 2021.

[323] *OpenSCHC*. https://openschc.github.io/openschc.

[324] Ali Ghodsi et al. "Information-centric networking: seeing the forest for the trees". In: *Proceedings of the 10th ACM Workshop on Hot Topics in Networks - HotNets '11*. ACM Press, 2011, 1–6. ISBN: 978-1-4503-1059-8. DOI: 10.1145/2070562.2070563. URL: http://dl.acm.org/citation.cfm?doid=2070562.2070563.

[325] Rehmat Ullah, Syed Hassan Ahmed, and Byung-Seo Kim. "Information-Centric Networking With Edge Computing for IoT: Research Challenges and Future Directions". In: *IEEE Access* 6 (2018), 73465–73488. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2884536.

[326] LoRa Alliance, Inc. *LoRaWAN Specifications*. https://lora-alliance.org/about-lorawan. Oct. 2019.

[327] *RIPE Atlas*. https://atlas.ripe.net/. 2019.

[328] Carles Gomez and Jon Crowcroft. "RTO considerations in LPWAN". In: *IETF lpwan working group* (July 4, 2019). https://www.ietf.org/id/draft-gomez-lpwan-rto-considerations-01.txt.

[329] Z. Shelby, K. Hartke, and C. Bormann. "The Constrained Application Protocol (CoAP) - RFC 7252". In: *IETF core working group* (June 2014). https://www.rfc-editor.org/info/rfc7252. DOI: 10.17487/RFC7252.

[330] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[331] C. Bormann et al. "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed - RFC 3095". In: *IETF lpwan working group* (July 2001). https://www.rfc-editor.org/info/rfc3095. DOI: 10.17487/RFC3095.

[332] Akhil Kumar, Vithala R. Rao, and Harsh Soni. "An empirical comparison of neural network and logistic regression models". In: *Marketing Letters* 6.4 (Oct. 1995), 251–263. ISSN: 1573-059X. DOI: 10.1007/BF00996189.

[333] Zheng Zhao et al. "LSTM network: a deep learning approach for short-term traffic forecast". In: *IET Intelligent Transport Systems* 11.2 (2017), pp. 68–75.

[334] A. Gensler et al. "Deep Learning for solar power forecasting — An approach using AutoEncoder and LSTM Neural Networks". In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2016, pp. 002858–002865. DOI: 10.1109/SMC.2016.7844673.

[335] Aicha Dridi et al. "An Artificial Intelligence Approach for Time Series Next Generation Applications". In: 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148931.

[336] *STM32L476 Microcontroller*. https://www.st.com/en/microcontrollers-microprocessors/stm32l476rg.html. 2021.

[337] *Semtech's SX1276MB1MAS*. https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276mb1mas. 2021.
[338] *STM32 Nucleo Expansion Board*. https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/stm32-nucleo-expansion-boards/x-nucleo-lpm01a.html. 2021.
[339] *EdgeImpulse*. https://edgeimpulse.com/. 2020.
[340] *Understanding LSTM Netwoks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. August 2, 2015.
[341] *TensorFlow Lite*. https://www.tensorflow.org/lite/. 2020.
[342] *TensorFlow*. https://www.tensorflow.org/. 2020.
[343] *C Implementation of Simple LSTM network*. https://github.com/Bernard-A/LSTM-by-Hand-FairyOnIce/blob/main/CPP/main.cpp. 2021.
[344] *Extract weights from Keras's LSTM and calcualte hidden and cell states*. https://fairyonice.github.io/Extract-weights-from-Keras's-LSTM-and-calcualte-hidden-and-cell-states.html. 2018.
[345] Hasim Sak, Andrew W Senior, and Françoise Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: (2014).
[346] *STM32 Cube Monitor Desktop Application*. https://www.st.com/en/development-tools/stm32cubemonitor.html. 2021.
[347] *The Things Network*. https://www.thethingsnetwork.org/. 2021.
[348] *Packet Broker*. https://www.packetbroker.org/. 2020.
[349] *IoTRoam Architecture*. https://github.com/afnic/IoTRoam-Tutorial/blob/master/Architecture.md. 2020.
[350] Bart Moons et al. "Using SCHC for an optimized protocol stack in multi-modal LPWAN solutions". In: *WF-IoT2019, the IEEE World Forum on Internet of Things*. 2019, pp. 1–6. DOI: 10.1109/WF-IoT.2019.8767210.
[351] *A Solution for Successful Interoperability with DLMS/COSEM and LoRaWAN*. https://lora-alliance.org/sites/default/files/2019-11/dlms-lorawan-whitepaper_v1.pdf. November 2019.
[352] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. "In Depth Performance Evaluation of LTE-M for M2M Communications". In: *The 12th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob'16*. New York, United States, Oct. 2016. URL: https://hal.archives-ouvertes.fr/hal-01451229.
[353] Joseph Azar et al. "An energy efficient IoT data compression approach for edge machine learning". In: *Future Generation Computer Systems* 96 (2019), pp. 168–175.

ECOLE
DOCTORALE

Title : Solving Interoperability and Performance Challenges over heterogeneous IoT Networks – DNS-based solutions

Keywords : Internet of Things, DNS, Interoperability, Compression, Roaming, Machine Learning

Abstract : The Internet of Things (IoT) evolved from its theoretical possibility to connect anything and everything to an ever-increasing market of goods and services. Its underlying technologies diversified and IoT now encompasses various communication technologies ranging from short-range technologies as Bluetooth, medium-range technologies such as Zigbee and long-range technologies such as Long Range Wide Area Network.
IoT systems are usually built around closed, siloed infrastructures. Developing interoperability between these closed silos is crucial for IoT use-cases such as Smart Cities. Working on this subject at the application level is a first step that directly evolved from current practice regarding data collection and analysis in the context of the development of Big Data. However, building bridges at the network level would enable easier interconnection between infrastructures and facilitate seamless transitions between IoT technologies to improve coverage at low cost.
The Domain Name System (DNS) basically developed to translate human-friendly computer host-names on a network into their corresponding IP addresses is a known interoperability facilitator on the Internet. It is one of the oldest systems deployed on the Internet and was developed to support the Internet infrastructure's growth at the end of the 80s. Despite its old age, it remains a core service on the Internet and many changes from its initial specifications are still in progress, as proven by the increasing number of new suggestions to modify its standard.
DNS relies on simple principles, but its evolution since its first developments allowed to build complex systems using its many configuration possibilities. This thesis investigates possible improvements to IoT services and infrastructures. Our key problem can be formulated as follow: Can the DNS and its infrastructure serve as a good baseline to support IoT evolution as it accompanied the evolution of the Internet?
We address this question with three approaches. We begin by experimenting with a federated roaming model IoT networks exploiting the strengths of the DNS infrastructure and its security extensions to improve interoperability, end-to-end security and optimize back-end communications. Its goal is to propose seamless transitions between networks based on information stored on the DNS infrastructure. We explore the issues behind DNS and application response times, and how to limit its impact on constrained exchanges between end devices and radio gateways studying DNS prefetching scenarios in a city mobility context. Our second subject of interest consists of studying how DNS can be used to develop availability, interoperability and scalability in compression protocols for IoT. Furthermore, we experimented around compression paradigms and traffic minimization by implementing machine learning algorithms onto sensors and monitoring important system parameters, particularly transmission performance and energy efficiency.

Titre : Contributions à la résolution de problèmes de performances et d'interopérabilité des réseaux IoT hétérogènes par l'utilisation du standard ouvert DNS et de services d'infrastructure

Mots clés : Internet des Objets, DNS, Interoperability, Compression, Itinérance, Apprentissage Machine

Résumé : L'Internet des Objets (IdO) a évolué depuis cette possibilité théorique de connecter tous les appareils à un réel marché de biens et de services en constante expansion. Les technologies sous-jacentes ont évolué et l'IdO repose aujourd'hui sur de nombreuses technologies de communication différentes: Des technologies à courte portée comme Bluetooth, moyenne portée comme Zigbee ou longue portée comme la technologie LoRa (Long-Range).
Les systèmes de l'IdO sont habituellement construits autour d'infrastructures fermées basées sur des systèmes en silo. Créer de l'interopérabilité entre ces silos fermés est un enjeu pour certains cas d'usages cruciaux dans le déploiement des technologies de l'IdO comme les villes intelligentes. Développer la problématique au niveau applicatif est une première étape directement inspirée des pratiques courantes en matière de collecte et d'analyse de données dans le cadre du développement des technologies de traitement de données massives. Cependant, construire des ponts au niveau réseau permettrait de faciliter l'interconnexion entre infrastructures et faciliterait la transition fluide entre technologies de l'IdO afin d'améliorer à bas coût la couverture réseau.
Le Système de Nom de Domaine (Domain Name System, DNS), initialement développé pour traduire les noms, lisibles et compréhensibles par les utilisateurs en adresses IP, utilisées par les appareils connectés, est reconnu comme un facilitateur sur les question d'interopérabilité sur Internet. C'est l'un des systèmes les plus anciens déployés sur Internet, développé à la fin des années 1980 pour supporter la croissance de l'infrastructures Internet. Bien qu'ayant beaucoup évolué ces dernières années, en témoignent les nombreuses propositions de modifications au standard publié à son sujet, le DNS reste aujourd'hui l'une des infrastructures les plus centrales du réseau Internet.
Le DNS repose sur des principes simples, mais son évolution depuis ses premiers développements ont permis de construire des systèmes complexes grâce à ses nombreuses possibilités de configuration. Dans le cadre cette thèse, qui étudie les possibles améliorations aux services et infrastructures de l'IdO, nous étudions la problématique suivante : Le DNS et son infrastructure peuvent-ils servir de support efficace à l'évolution de l'IdO de la même manière qu'il a accompagné l'évolution d'Internet ?
Dans cette optique, nous étudions de possibles améliorations de systèmes de l'IdO sous trois angles. Nous testons tout d'abord un modèle d'itinérance pour réseaux de l'Internet des Objets au travers de la construction d'une fédération reposant sur l'infrastructure du DNS et ses extensions pour en assurer l'interopérabilité, la sécurité de bout-en-bout et optimiser les communications entre infrastructures. Son objectif est de proposer des transitions fluides entre réseaux sur base d'informations stockées à l'aide de l'infrastructure DNS. Nous explorons également les problématiques introduites par le DNS, notamment en termes de latence et d'influence sur les temps de réponse des applications, et comment en limiter l'impact sur les échanges, déjà grandement contraints, entre objet connecté et passerelle radio. Pour cela nous étudions les conséquences de l'utilisation de requêtes DNS anticipées dans un contexte de mobilité en milieu urbain. Nous étudions ensuite comment le Système de Nom de Domaine peut renforcer l'interopérabilité, la disponibilité de ressources et le passage à l'échelle de systèmes de compression de paquets de l'IdO. Enfin, nous explorons la question de la minimisation de trafic en implantant des algorithmes d'apprentissage sur des capteurs et en mesurant les paramètres du système final, en particulier en terme de performances de transmissions et d'efficacité énergétique.

Institut Polytechnique de Paris
91120 Palaiseau, France