THESE

en vue de l'obtention du grade de Docteur, délivré par

l'ECOLE NORMALE SUPERIEURE DE LYON

Ecole Doctorale N° 512
Informatique et Mathématique de Lyon (Infomaths)

Discipline : Informatique

Soutenue publiquement le 04/10/2023, par :

Samir SI-MOHAMMED

Multi-Criteria Selection and Configuration of IoT Network Technologies

Sélection et Configuration Multicritères de Technologies Réseau IoT

Devant le jury composé de :

THÉOLEYRE Fabrice	Directeur de Recherche	CNRS	Rapporteur
WATTEYNE Thomas	Directeur de Recherche	Inria	Rapporteur
GUITTON Alexandre	Professeur	Université de Clermont Auvergne	Examinateur
NICOARA Angela	Professeure	Lucerne University of Applied Sciences and Arts	Examinatrice
BEGIN Thomas	Professeur	Université Claude Bernard Lyon 1	Directeur de thèse
GUÉRIN LASSOUS Isabelle	Professeure	Université Claude Bernard Lyon 1	Co-directrice de thèse
VICAT-BLANC Pascale	Directrice de Recherche	Inria	Co-directrice de thèse

THÈSE DE DOCTORAT

Opérée au sein de

l'École Normale Supérieure de Lyon

Discipline : **Informatique**

École Doctorale 512 InfoMaths

Thèse préparée par

Samir Si-Mohammed

Multi-criteria Selection and Configuration of IoT Network Technologies

---

Devant le jury composé de :

M. Fabrice Theoleyre *Directeur de Recherche, CNRS*	**Rapporteur**
M. Thomas Watteyne *Directeur de Recherche, Inria*	**Rapporteur**
M. Alexandre Guitton *Professeur, Université Clermont Auvergne*	**Examinateur**
Mme Angela Nicoara *Professeure, Lucerne University of Applied Sciences and Arts*	**Examinatrice**
M. Thomas Begin *Professeur, Université Claude Bernard Lyon 1*	**Directeur de Thèse**
Mme Isabelle Guérin Lassous *Professeure, Université Claude Bernard Lyon 1*	**Co-Directrice de Thèse**
Mme Pascale Vicat-Blanc *Directrice de Recherche, Inria*	**Co-Directrice de Thèse**

Acknowledgments

Il me serait difficile de remercier nommément toutes celles et ceux qui, d'une manière ou d'une autre, m'ont aidé à mener à bien ce travail de thèse. Puissent-elles et puissent-ils trouver ici, l'expression de ma profonde reconnaissance.

J'adresse en tout premier lieu mes plus sincères remerciements à l'ensemble des membres de mon Jury, pour l'honneur qu'ils me font en acceptant d'examiner ce travail : M. le Professeur Alexandre GUITTON, pour avoir accepté d'en assurer la présidence, MM. les Professeurs Fabrice THÉOLEYRE et Thomas WATTEYNE, qui ont accepté d'en être les rapporteurs, pour le temps qu'ils ont consacré à la lecture de mon manuscrit et pour leurs précieux retours, ainsi que Mme la Professeure Angela NICOARA qui a accepté de participer à ce jury en tant qu'examinatrice. Je leur exprime ici toute ma gratitude.

J'exprime également ma plus profonde gratitude à celles et ceux qui m'ont accordé leur confiance, d'abord en me confiant ce travail, et ensuite pour leur soutien et leur appui tout au long de son déroulement. Thomas, merci pour tout, pour ton aide, pour tout le temps que tu as passé à me relire, pour tous tes conseils, ainsi que pour ta rigueur grâce à laquelle j'ai tant appris. Isabelle, pour ta gentillesse, ton soutien permanent, ton aide précieuse, tant sur les aspects scientifiques que sur les nombreux aspects administratifs qu'il a fallu gérer, ainsi que tes nombreux encouragements qui m'ont permis d'améliorer sans cesse mon travail. Enfin, Pascale, très simplement, merci pour tout. Merci pour Stackeo, sans laquelle cette thèse n'aurait pas pu voir le jour. Merci pour ton implication permanente et si précieuse dans ce travail ainsi que dans la rédaction de mon manuscrit, merci pour le niveau d'exigence que tu m'as toujours imposé, et qui m'a fait donner le meilleur de moi-même, même si je crois pouvoir avouer aujourd'hui qu'il m'est arrivé de ne le comprendre qu'a posteriori. En un mot, et du fond du coeur, merci d'avoir été là chaque fois que j'en ai eu besoin. Thomas, Isabelle et Pascale, vous m'avez tous trois, sur tous les plans, permis d'effectuer ma thèse dans les meilleures conditions possibles, merci de croire à mon infinie reconnaissance.

Je tiens à remercier Mme la Professeure Catherine Rosenberg, de l'Université Waterloo au Canada, qui a accepté de m'accueillir au sein de son équipe durant trois mois, dans le cadre d'une mobilité internationale. Travailler avec toi a été une expérience marquante et profondément enrichissante. Au delà des échanges fructueux que j'ai pu avoir avec toi, ta disponibilité à mon égard et la chaleur de ton accueil m'ont profondément touché. Mon séjour à Waterloo fut un réel plaisir, et je tiens à t'adresser ici mes plus sincères remerciements.

Je remercie également ceux dont j'ai partagé le quotidien durant ces trois dernières années, l'ensemble des collègues d'HoWNet mais aussi d'OCKHAM : Clément, Tung, Léon, Esther, Antoine, Badr, Titouan, Mathurin, Guillaume, Meriem, Can, Ayoub, Myriam, Valérie, Théotime, Johann, Youssouph et enfin Mathieu avec qui j'ai eu la chance de participer à ma première conférence en présentiel, et pour laquelle je garderai un très bon souvenir. Enfin, une pensée très chaleureuse à mon co-bureau Anthony, merci pour ta complicité, ta bonne humeur, et tout simplement merci pour ton amitié. Je suis ravi d'avoir pu partager cette aventure avec toi. Venir au bureau quotidiennement était un réel plaisir pour moi, et tu y as grandement contribué.

Je tiens à remercier également les collègues de STACKEO : Fabien, Guilhem, Antoine, Anthony, Zakaria, Charlotte, José, Robin. Merci pour votre accueil chaleureux, ainsi que pour tous les moments que nous avons partagés, notamment nos fameux "fun time" des Vendredi après-midi ! Merci également à toute l'équipe du MALIP : Marie Narducci, Marie Bozo, Chiraz et Solène pour leur disponibilité et pour leur aide.

I would like to thank all the team at Waterloo: Maryam, Ahmad, Emam, Paulo, Euan, Mohammad, Haseen, Kiril, Abdallah and Ara. Thank you so much for all the moments we spent together and for the warm welcome, I really enjoyed my time there thanks to you. I would also like to thank the Hedrick family (including Stella and Ollie) for their warm welcome as well, and for their kindness during my stay.

J'en profite aussi pour remercier Mohamed FRENDI qui a grandement contribué à mon séjour à Waterloo, merci pour ton accueil extrêmement chaleureux, pour ta gentillesse ainsi que pour tous ces merveilleux moments passés ensemble.

Une pensée particulière à tous mes amis du Relais Yves Chauvin avec qui j'ai eu la chance de partager de nombreux moments : Hamza, Assil, Aghiles, Nabil, Mohamed, Imad, Mehdi, Rasha, Riad, Jean-Eudes, Angélique, Coralie, Jawad, Ikram, Marine. Merci à vous tous, j'ai grandement apprécié votre compagnie.

Bien-sûr, je souhaiterais remercier très chaleureusement ceux qui furent ma deuxième famille, Yacine, Brahim et tout particulièrement Raouf. Merci pour tous les services inestimables que vous m'avez rendu, et surtout d'avoir toujours été là pour moi. J'ai eu la chance de vous avoir à mes côtés, tout en sachant que je pourrais compter sur vous en toutes circonstances. J'aimerais aussi remercier tous ceux qui m'ont soutenu jour après jour, malgré la distance : Feriel, Haqou, Khaled, Léna, Sara, Lynda, Abdessalem, Saloua, Imane et tant d'autres. Merci pour votre amitié, pour votre soutien et d'avoir toujours cru en moi.

Enfin, je souhaiterais remercier ceux sans qui je n'en serais pas là aujourd'hui. Ma chère mère, pour sa tendresse, pour sa douceur, et d'avoir toujours veillé à ce que je prenne soin de moi avant tout. Ma belle-soeur Manel, pour sa gentillesse, sa générosité, sa bonne humeur ainsi que ses petits plats qui n'étaient pas loin d'être la raison principale de mes déplacement fréquents à Lille, merci pour tout. Mon grand-frère Hakim, pour son affection, son soutien permanent, ses précieux conseils, et surtout de me rendre si fier, lui qui ne cesse de me mettre la barre de plus en plus haut. Comme le dit si bien un dicton Kabyle : "Win ur nes3i tagmatt mehqur", mais pour ma part, je n'ai pas à m'en faire. Enfin, mon cher père, merci pour absolument tout, merci d'être mon plus grand soutien, de n'avoir jamais douté de mes capacités, de m'avoir toujours poussé à me surpasser et à toujours viser plus haut. Merci aussi d'être mon plus grand modèle, tant sur le plan professionnel que personnel. Merci à ce fameux CDC, d'avoir tout simplement fait moi celui que je suis aujourd'hui.

Voilà, je m'excuse envers toutes celles et ceux qui auraient contribué de près ou de loin à ce travail, et que j'aurais inconsciemment oublié de mentionner. Qu'ils sachent que ma gratitude leur est acquise.

Abstract

The Internet of Things (IoT) is generally defined as a global infrastructure that enables advanced services by interconnecting physical and virtual things based on interoperable network technologies. It provides access to a wide range of applications, such as smart cities, smart agriculture, and asset tracking. The increase in the number and diversity of network technologies brings new challenges to IoT architects, who are in charge of the design and deployment of IoT solutions. Indeed, the selection and configuration of network technologies are crucial to ensure the good behavior of IoT applications. In this thesis, we address this issue through the development of a set of methods and associated tools that help IoT architects in their process of designing and deploying IoT solutions. First, we propose a generic framework for the performance evaluation of an IoT network technology given a specific application context. Second, we introduce a new algorithm relying on multi-criteria optimization methods that enables the automatic selection of the most adapted network technology for a given application context. Then, we address two limits of our method which are cost (in terms of time) and accuracy. For cost, we propose a solution for accelerating the design decision using simulation models and regression methods, with an application on the configuration optimization of IoT network technologies. For accuracy, we explore the possibility of calibrating simulation models with data collected from a real deployment using a small-scale Proof of Concept (PoC). Finally, we implemented these methods in a no-code tool, integrated into the Stackilab platform, relying on simulation and aiming to help IoT architects easily future-proof an IoT solution.

Contents

List of Figures

List of Tables

Introduction 1

1.1 Context

There are many definitions for the Internet of Things (IoT). The International Telecommunications Union (ITU) defines it as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and network technologies" [1]. According to Wikipedia*, it is a concept that describes "physical objects (or groups of such objects) with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.". Embedded in various industries, organizations, homes, etc., IoT has become an indispensable and seamless part of our modern existence. IoT devices are omnipresent, serving diverse purposes. For instance, asset trackers provide real-time location updates and notify us about potential delivery delays, while smart meters monitor energy usage. Sensors detect water leaks and air pollution, and remote control systems automate manufacturing processes. These devices enable the development of innovative applications and services that leverage real-time data and remote control capabilities. This convergence of the physical and virtual realms has the potential to enhance operational efficiency, promote sustainability, and safeguard the well-being of individuals and the environment.

[1]: Biggs et al. (2016), 'Harnessing the Internet of Things for Global Development'

In order to meet the diverse requirements of the use-cases, IoT solutions are deployed in different sectors. An IoT solution refers to a set of hardware, software and network technologies to enable the connection of physical devices (sensors, actuators, etc.) to the Internet. We distinguish two types of IoT solutions: (i) Pre-packaged and (ii) tailored. A pre-packaged IoT solution typically includes pre-configured sensors or devices, connectivity modules, data management and analytics software, etc. These solutions are developed by IoT vendors or providers who package the necessary components and capabilities into a single solution, reducing the complexity and time involved in building a custom IoT system from scratch. A tailored IoT solution refers to a customized implementation of IoT technologies and components to meet the specific requirements of an organization or industry. Unlike pre-packaged solutions, which offer a standardized set of components, a tailored IoT solution is designed and developed specifically for a particular use-case or business need.

A wide range of technologies are readily available for the implementation of end-to-end IoT solutions. Network technologies play a crucial role in connecting the various subsystems that constitute the complete IoT communication system, following the ISO/OSI model. At the interface of this interconnection, the physical IoT network serves as a critical subsystem, facilitating the connectivity of IoT devices to the Internet. Moreover, network technologies may determine the viability of an IoT

* https://en.wikipedia.org/wiki/Internet_of_things

solution, in terms of Quality of Service (QoS), but also in terms of economic profit. IoT end-devices are known to be small components that are supposed to have long lifetimes. Ensuring a low energy consumption at the network level is therefore a priority. From relatively old and established network technologies (such as Wi-Fi or Bluetooth) to much more recent ones (LoRaWAN, 5G, etc.), the technological offer in terms of data communication through the use of the radio spectrum has increased considerably, both in quantity and in quality. It is this diversification, as well as the improvement in performance that has accompanied it, that has propelled IoT to the forefront in terms of research and development, making it almost impossible today to find a sector that has not been impacted in some way by the IoT.

[2]: Lantronix Inc. (2020), 'Product Life Cycle in the Age of IoT'

The lifecycle of an IoT solution consists of five phases [2]: (i) Design, which consists of the design of the system and the service components, (ii) Development, which involves developing software and hardware components, (iii) deployment, where the devices are deployed and their services (data collection, etc.) installed, (iv) operations, where the services are started and ready to be used and (v) decommissioning, where the service is decommissioned and the solution terminated. Tailored and pre-packaged IoT solutions have distinct lifecycles. Tailored solutions involve extensive customization to meet specific project requirements, requiring design and development from scratch. This induces longer development time and higher complexity. On the other hand, pre-packaged solutions are ready-made, commercially available options with predefined features. They can be implemented quickly with simplified deployment and lower complexity.

[3]: Badnakhe (2022), 'A Blueprint on IoT Solutions Development'

In addition, according to [3], the lifecycle of an IoT solution requires the presence of several stakeholders, such as (i) IoT architects, (ii) IoT developers, (iii) data analysts and (iv) security architects. In order to understand the role of each stakeholder, let us use a smart building example. We consider the case of the instrumentation of a commercial building already equipped with a Building Management System (BMS) in charge of the remote control of Heating, ventilation and air conditiong (HVAC) systems as well as water and energy regulation. The facility manager wants to add a tailored smart solution to finely monitor the building, gain better visibility on its real usage and to better adapt the building services. For this, the facility manager would like to deploy a range of sensors: Entrance detectors, occupancy monitors, air quality sensors, temperature sensors, smart lighting and other end-devices. During the design phase, IoT architects are responsible for designing the overall architecture of the IoT solution, including the hardware and software components, as well as the communication protocols used to connect the devices. IoT developers are responsible for developing the software applications and firmware for the IoT end-devices that will be deployed within the smart building. Then, the data analysts are responsible for analyzing the data collected by the IoT end-devices to identify patterns and trends that can be used to optimize the smart building. The security architects are here to ensure that the IoT solution is secure and protected against unauthorized access. During the operations phase, the different stakeholders make sure that the solution is behaving correctly: IoT architects monitor the performance of the system to see if the desired objectives of performance are attained (they can trigger changes in the architecture in deemed necessary), IoT developers maintain and update the firmware of the end-devices if needed, data analysts collect and identify patterns and anomalies in the data and security architects ensure that the system is still secure through conducting regular security audits. Finally, the solution is terminated with the decommissioning phase.

IoT solutions can be deployed in countless domains, featuring a multitude of use-cases (or applications): Radiation monitoring, where sensors are used to measure the level of radiation in a nuclear facility, video-surveillance, where video streams are used to monitor an event, or smart home, where we can have near-real-time monitoring of the temperature or brightness of the rooms in a house, are just some examples. IoT use-cases/applications are mainly characterized by their traffic workload (message size and period, etc.). One of their specifics is that the majority of IoT use-cases and applications evolve over time [4]. The traffic workload often evolves with time, for instance, an increase in message frequency. These evolutions typically trigger adaptations in the IoT solution during the operations phase of its lifecycle.

[4]: Udoh et al. (2018), 'Developing IoT Applications: Challenges and Frameworks'

IoT Economic Impact

The deployment of IoT solutions in different domains is highly driven by its economic impact. According to McKinsey[†], IoT solutions will have a total economic impact of 3.9 trillion USD to 11.1 trillion USD per year in 2025. Accenture [5] states that the Industrial IoT has the potential to add 14.2 trillion to the global economy by 2030, with manufacturing and healthcare being the two industries with the largest potential values. Moreover, tens of new connected products and pre-packaged or tailored solutions are launched every day. Analysts predict that by 2030, the IoT could enable from 5.5 to 12.6 trillion dollars in value globally, including the value captured by consumers and customers of IoT products and services [6]. For example, shipments of asset trackers will grow by more than 50 percent annually through 2024 [7].

[6]: Chui et al. (2021), 'The Internet of Things: Catching up to an Accelerating Opportunity'

[7]: ABI research (2020), 'IoT Asset Tracking Device Evolution: Technologies and Features Driving the Next Stage of Growth'

This economic impact can be explained by the potential of IoT for cost savings through improved efficiency. For example, smart sensors can be used to monitor energy consumption, water usage, and other resources in real-time, allowing companies to optimize their operations and reduce waste. Also, the ability to automate processes can lead to increased productivity. Moreover, new businesses are emerging thanks to IoT. For instance, companies can offer new services based on data collected from connected devices, or use IoT to develop new products and services that meet emerging customer needs. Finally, existing businesses can be enhanced with measured user data. As an illustration, IoT sensors can be used to collect data on customer behavior, such as how they interact with products and services. This information can be used to customize marketing messages and improve customer engagement. All these facets can be the reason for an improved Return on Investment (ROI) of businesses relying on IoT.

[†] https://www.mckinsey.com/mgi/overview/in-the-news/by-2025-internet-of-things-applications-could-have-11-trillion-impact

IoT Development Obstacles

These potential economic advances have encouraged an impressive amount of effort toward the improvement of IoT solutions, at all levels. Usually, an IoT solution is divided into three major components [8]: (i) Physical devices, (ii) network technologies and (iii) high-level applications. Every component within an IoT solution has the potential for improvement through the application of scientific approaches. For instance, building less energy-intensive devices is an important possible improvement. Also, proposing and optimizing wireless network technologies to allow more efficient communication can be of great interest. Finally, innovations must also step in at the application layer. Indeed, it is necessary to propose new schemes for data management and data mining, as, on the one hand, more storage is needed to deal with the heterogeneous nature and important volume of personal and enterprise data, and on the other hand, traditional data mining techniques are not directly applicable to unstructured images and video data [9].

[8]: Wu et al. (2010), 'Research on the Architecture of Internet of Things'

[9]: Lee et al. (2015), 'The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises'

[10]: Kassab et al. (2020), 'A–Z survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations'

According to [10], the networking component of IoT is the layer that attracted the most interest among researchers from 2010 to 2020. However, an important issue is that the widespread use of network technologies, particularly on an industrial scale, despite all its advantages in terms of offer, is generating new problems of interoperability. In fact, the more widespread the use of these IoT network technologies becomes, the more numerous and diverse the parameters to be taken into account. In addition, this also causes new issues at stake, including the need to adapt the network technologies chosen or the architectures selected not only to a globally constrained environment but also to specific needs in a particular application domain. Since there are several applications that can be targeted in IoT, we believe that it will soon no longer be appropriate for IoT researchers to propose methods and solutions dedicated to a given network technology or for a specific application context. There is a crucial need of bringing more abstract solutions that can be generalized to any context and adapted to other solutions.

[11]: Baccelli (2021), 'Internet of Things (IoT): Societal Challenges & Scientific Research Fields for IoT'

Besides that, traditionally, research conducted in the IoT domains targets too much an improvement in raw performance (data rate, accuracy, etc.) while neglecting the potential negative impact of such improvements [11]. Indeed, the quest for only more performance is no longer affordable in the current climate context: Environmental impact must be carefully considered when proposing improvements to IoT network technologies. This need is even more tangible when the yielded performance improvement is not absolutely mandatory. The current environmental challenge humanity is facing makes us sincerely think that we need to question our relationship to science and that more performance at any cost is not an option, although this can seem paradoxical since we also believe that the sake of humanity resides in science. To put it in a few words, we believe that for researchers in any domain, and most particularly in IoT, the environmental impact must be incorporated in every solution or method they propose.

IoT Adoption Obstacles

Beyond the fact of providing users with a diverse and varied technological offer, the multiplication of network technologies leads to the problem of choosing one (or several) network technology(ies) among all those that could potentially meet the need while being adapted to the targeted application domain. Indeed, it is agreed that there is no one-fits-all solution [12]. Therefore, obtaining the best compromise in terms of performance and energy consumption, while taking into account costs and execution times is an absolute necessity. Without forgetting the need to guarantee interoperability between technologies with different designs and objectives, which must sometimes coexist in the development of the solution to a specific need. Finally, the selection of a network technology must also be motivated by the scalability of the solution, *i.e.*, its ability to support changes in the application workload or even in the number of nodes. It is absolutely necessary that the deployed IoT solution can scale, otherwise, it would lead to a disastrous waste of time, money and energy.

[12]: Vannieuwenborg et al. (2018), 'Choosing IoT-connectivity? A guiding Methodology based on Functional Characteristics and Economic Considerations'

In addition to that, the complexity of evaluating the relevance of an IoT network technology for a given application is a real challenge for IoT architects. Indeed, as observed by the community of researchers, the link between the performance of an IoT network technology and the targeted application may yield the need for a holistic approach to the evaluation. In other words, the evaluation must take into account the different parameters that can affect the performance of an IoT network technology, as well as their interdependencies. To perform this evaluation, there is the need to model the targeted IoT application in a way that includes very varied aspects, whether they are topological aspects (number of nodes, positions, etc.), aspects related to traffic (size and periodicity of data transmission), related to the radio environment (presence of interference, etc.) or other parameters. One of the major obstacles faced is precisely the multiplicity of these aspects, each of which can be more or less decisive, depending on the actual case of the study.

This complexity may confuse IoT architects. Making decisions in such a context can be complex and risky, with potentially far-reaching consequences for the success of the IoT solution. Choosing an unadapted network technology for an IoT solution is not a viable option due to budget, capacity, and performance constraints. Gaining visibility into all the factors impacting the performance and energy consumption of an IoT solution can be very challenging, and making accurate forecasts can be difficult, time-consuming, and prone to error if not performed carefully. In addition, IoT architects often do not have the time nor the networking skills to perform thorough testing. The concepts of performance evaluation of IoT network technologies may therefore be inaccessible to some of them, which can slow down the deployment of their solution. There is therefore a need to "democratize" the rigorous analysis of IoT network technologies, in the sense of offering them the means to carry out such evaluations, in particular by providing them with analysis tools, so that they can take advantage of them without increasing the complexity of their task.

1.2 Research Issues

In this thesis, we propose to address the two following issues:

Selecting an IoT Network Technology

The first issue deals with the choice of an IoT network technology for a pre-packaged or a tailored IoT solution, according to the application domain and based on the technical characteristics that must be taken into account as well as the application requirements. These considerations illustrate the need to solve a problem that can be considered as transversal, regardless of the network technology(ies) considered, the chosen application domain and the specific requirements of the user application. The question is: **How to make comparisons between different potential network technologies in order to allow the selection of the one(s) that will best meet the requirements?** This necessarily requires being able to evaluate each of the applicable network technologies, while taking into account its characteristics, and considering specific needs in a particular field of application, *i.e.*, Quality of Service (QoS). This process of selection is crucial since it takes place during the design phase of an IoT solution lifecycle, and can therefore lead to dramatic effects if not done carefully. Indeed, such a selection must incorporate the fact that the evaluation of an IoT network technology will not always focus on the same parameters and may vary greatly depending on the application. For example, for a radiation monitoring application, it is critical that the measured level of radiation is received correctly (i.e., avoiding major losses), in order to take appropriate decisions about isolating the irradiated area. In this case, only minimal data losses could be acceptable in the sense that they would not necessarily compromise the validity of the results in terms of usability in decision-making, while the time packets take to be received is not a problem. In comparison, in the case of video-surveillance, it might be acceptable for some video frames to be lost, which might deteriorate the quality of the video, but not its usefulness, while it will be essential that the video streams are imperatively received in a very short time, in order to allow for real-time decision-making in case of emergency.

Configuring an IoT Network Technology

The second issue is about the configuration of a given network technology. Indeed, each IoT network technology can be configured differently, according to many configuration parameters (*e.g.*, spreading factor[1], coding rate[2] or type of traffic[3] for LoRa). These parameters can take several values, with a potentially significant impact on the performance obtained. It is crucial that the configuration decision is carefully made. Indeed, it can strongly impact the solution's performance during the operations phase. Moreover, a re-configuration can be triggered during that phase if deemed necessary *e.g.*, the application characteristics evolve over time, it can cause a waste of time and money. In addition to this considerable space of possible configurations, the performance will strongly depend on the targeted application, as the same configuration can be efficient for one application and much less efficient for another. This motivates the interest in relying on the finest characteristics of the

1: This parameter determines the speed at which the signal frequency changes across the bandwidth of a channel.

2: An indication of how much of the data stream is actually being used to transmit user data.

3: Determines whether the data is sent with or without an acknowledge.

application, and of the configuration parameters considered here, in order to anticipate and guide the choice of such or such other configuration. This choice should be made according to the desired application, in view of the important improvements (or deterioration) that it can generate in the performances.

To address the selection and configuration issues, two aspects appear to be essential to be considered: First, the type of approach implemented for the performance evaluation, and second, the evaluation metrics to take into account.

Performance Evaluation Approaches: Regarding the performance evaluation, it can be done in three different ways:

- Experimentation, which consists in conducting real experiments using real material/devices using "hardware probes and software probes. Hardware probes are entities that are connected to the hardware devices being measured. " [13],
- Analytical models, which are "mathematical models used to answer a specific question or make a specific design decision, and which must be expressed with sufficient precision that they can be formally analyzed, which is typically by a computer" [14],
- Simulation, which is the "process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the operation of the system" [15].

[13]: Heidelberger et al. (1984), 'Computer Performance Evaluation Methodology'

[14]: Friedenthal et al. (2014), *A Practical Guide to SysML: The Systems Modeling Language*

[15]: Shannon (1998), 'Introduction to the Art and Science of Simulation'

Each of these methods has its own advantages and disadvantages: Experimentation may allow to have precise and grounded results, but at a high cost because of the necessity of using the physical object. On the other hand, modeling can be done at a lower cost but can lack precision in its predictions of the behavior given the absence of the physical object. Simulation seems to be a good compromise between cost and accuracy, because it allows us to have a good overview of the performances while offering an easy configurability, and at a lower cost since there is no need to have the real hardware. However, simulation can also carry some burdens in terms of execution time or reliability.

As a side note, for the execution time, it would be interesting to explore if the outcome of previously executed simulations can be reused to predict the outcome of future ones. Machine Learning methods, which are known to quickly identify trends and patterns by simultaneously analyzing a large volume of data [16], seem to be relevant for this kind of problem. It would therefore be scientifically interesting to explore how simulation can be coupled with Machine Learning to deal with the execution time problem. Regarding the problem of reliability, one potential way of relieving it would be to combine simulation and experimentation to feed the simulator with real data and see whether it can allow the production of more grounded data. Thus, exploring the merits of coupling simulation and experimentation can be worth it.

[16]: Uddin et al. (2022), 'Machine Learning in Project Analytics: A Data-driven Framework and Case Study'

Performance Metrics: Turning now to performance metrics, it seems natural enough that each application domain may have one metric that is more important than others. However, this does not negate the potential impact of these other metrics on the performance achieved. For example, optimizing energy consumption would be a top priority for any

IoT architect deploying a battery-powered sensor-based solution, since they are very constrained devices. Nonetheless, it does not mean that optimizing this metric alone and ignoring other parameters would be the best solution. It is therefore more judicious to examine the possible relationships between different metrics to an optimal compromise.

The problem of selecting the most relevant network technology and its configuration is a multi-criteria optimization problem. To tackle it, it seems possible to reduce its level of complexity, starting with the examination of the inter-dependencies that may exist between performance metrics, and ensuring that they are compatible with the objectives that one wishes to achieve. It is unlikely, for example, to consider maximizing the amount of transmitted data without consuming more energy. Moreover, it is necessary to ensure that the methods implemented for the multi-criteria optimization will be adapted, by offering the capacity to take the probable non-convexity of the function to be optimized into account. Knowing that the cost of an IoT solution is determined by several parameters such as the equipment prices, the spectrum subscription fees, etc., it would be also interesting to examine its inter-dependency with the performance metrics as well as the energy consumption. Better performances don't necessarily come with higher costs. Moreover, it must be kept in mind that the energy consumption of an end-device impacts its battery lifetime, which in its turn determines how many times the battery should be replaced (maximal number of cycles), which finally causes additional costs.

1.3 Objectives

The main objective of this thesis is to explore possible solutions that allow IoT architects to make informed decisions in terms of IoT network technology to be deployed but also in terms of its configuration. Ideally, this would be through interactive and accessible tools allowing them to have all the necessary elements for decision-making, based on the appropriate performance evaluation metrics. We examine the possibility of coupling different evaluation tools (experimentation and simulation) to calibrate the simulation models to make them more realistic. More concretely, the global objective of this work consists in examining the possibility of achieving, through modeling and simulation, an analysis of the adequation of a technological configuration of the IoT network for a given application context.

1.4 Thesis Organization

This thesis is organized into eight chapters. Chapter 1 defines the context the objectives, and the underlying issues that we propose to address in this thesis. Then, we present in Chapter 2 a background with the main concepts related to our work. We present a brief history of the IoT as well as the scientific motivations behind the considerable growth of this discipline in recent years. We then focus on the main existing application domains of IoT. Then, we describe in a more formalized way IoT solutions and their architectures. After that, we give an overview of the different existing network technologies in the market. Finally, we present the various performance metrics used as a basis for the evaluation of IoT network technologies.

Chapter 3 is a state-of-the-art on the evaluation, selection, configuration and simulation of IoT network technologies. First, we provide a brief overview of the field approaches that IoT architects usually refer to for the evaluation of network technologies before moving to the deployment. Then, we present the scientific approaches that researchers have been working on so far. In that context, we present the different evaluation approaches that exist, before presenting some related work for the evaluation, the selection and the configuration of network technologies.

In Chapter 4, we present the first scientific contribution aiming to answer the following questions: **What are the main aspects to be considered for analyzing the adequacy of an IoT network technology with a given application? How can we propose an abstract evaluation process, applicable regardless of the network technology, the application, the metrics or the evaluation tool?** We propose a framework that consists of modeling IoT applications and network technologies in order to evaluate the relevance of an IoT network technology for a given application, according to different evaluation metrics and using any evaluation tool, with an emphasis at this stage on energy consumption.

Chapter 5 focuses on the following question: **How can we automate the process of the selection of the best network technology to consider for a given application context along with the potential scalability?** We propose a methodology for selecting the most efficient IoT network technology for an IoT application, based on the requirements of the targeted application domain. Our solution is based on a Multi Attibute Decision Making (MADM) approach, in which we take into account the impact of the topology on the performance as well as the scalability of the solution.

We focus on Chapter 6 on addressing two limitations of simulation for the proposed selection methodology presented in Chapter 5, which are cost and reliability. We focus on the two following questions: **How can we accelerate the methodology to propose an optimized configuration?** and **How can we use enhance the reliability and the accuracy of simulation models?** For the first question, we propose a generic method based on Machine Learning (regression methods) to optimize the configuration of a given network technology for a given application. For the second one, we propose a method for calibrating the simulation models from data gathered from real experimentation to have more reliable results, with a focus on energy consumption.

Finally, in Chapter 7, we tackle another limitation of simulation, which is accessibility. The question we propose to explore is **How can we allow IoT architects, which are not necessarily network experts nor programmers, to benefit from rigorous analysis for the selection of the best network technology during the design phase of their solution?** We propose a decision support tool, intended for IoT architects and capable of assisting them during the lifecycle of the solutions they plan to implement. Beyond the provided answer in terms of technological and/or configuration choices, the main interest of this tool lies in the fact that it allows any architect to have direct access to the network simulation without having

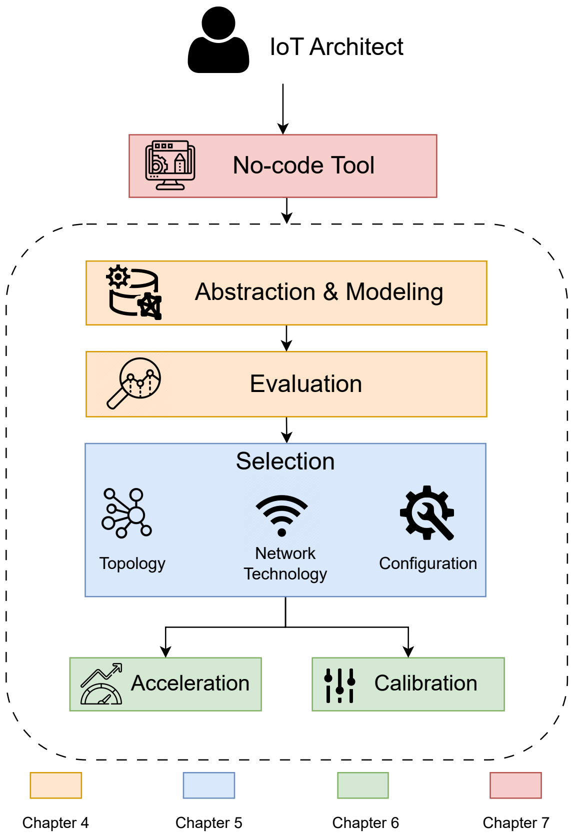


Figure 1.1: Thesis Contributions and Outline.

to resort to any programming activity, or even to have deep expertise in networks.

The conclusion is provided in the last chapter of this thesis.

Figure 1.1 depicts the contributions as well as the outline of this thesis. As we can see, the contributions of Chapter 4 are the basis for the contributions of Chapters 5, 6 and 7. Chapter 5 is the central contribution of the thesis, while Chapter 6 is addressing some of its limitations. Finally, the contributions of Chapter 7 represent the interface between all the remaining contributions and the IoT architect, to allow IoT architects to benefit from them in a seamless way.

1.5 Publications

International Journals

- Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. "HINTS: A Methodology for IoT Network Technology and Configuration Decision". In *IoT Journal*, volume 22. Elsevier, 2023.

- Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. "Smart Integration of Network Simulation in Network Digital Twin for Optimizing IoT networks". **Submitted** to *Future Generation Computer Systems*, Elsevier, August 2023.

International Conferences

- Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. "ADIperf: A Framework for Application-driven IoT Network Performance Evaluation." In *2022 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1-8. IEEE, 2022.

- Samir Si-Mohammed, Malasri Janumporn, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. "SIFRAN: evaluating IoT networks with a no-code framework based on ns-3." In *Proceedings of the 2022 Latin America Networking Conference (LANC)*, pp. 42-49. ACM, 2022.

- Samir Si-Mohammed, Zakaria Fraoui, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. "StackNet: IoT Network Simulation as a Service". In *2023 IEEE International Conference on Communications (ICC)*, pp. 1-6. IEEE, 2023.

National Conferences

- Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. "COSIMIA : Combiner Simulation et Apprentissage Automatique pour l'Optimisation des Configurations Réseau IoT", In *CoRes 2023 - 8èmes Rencontres Francophones sur la Conception de protocoles, l'évaluation de performances et l'expérimentation de Réseaux de communication*, pp 1-4. 2023.

Software

- SIFRAN, an online no-code tool for simulation IoT solutions in ns-3, `https://github.com/Stackeo-io/SIFRAN`.

Other Publications

The following works have been published outside this thesis.

- Samir Si-Mohammed, Maha Bouaziz, Hamed Hellaoui, Oussama Bekkouche, Adlen Ksentini, Tarik Taleb, Lechoslaw Tomaszewski, Thomas Lutz, Gokul Srinivasan, Tanel Jarvet, and Pawel Montowtt. "Supporting unmanned aerial vehicle services in 5G networks: New high-level architecture integrating 5G with U-space". In *IEEE Vehicular Technology Magazine*, Volume 16(1), pp. 57-65. IEEE, 2021.

- Samir Si-Mohammed, Adlen Ksentini, Maha Bouaziz, Yacine Challal, and Amar Balla. "UAV mission optimization in 5G: On reducing MEC service relocation". *2020 IEEE Global Communications Conference (Globecom)*, pp. 1-6. IEEE, 2020.

Background | 2

We present in this chapter a global overview of IoT, its evolution and history, before introducing the trendiest IoT applications. Then, we present some of the most famous IoT network technologies available in the market. We then examine different metrics used for the evaluation of IoT network technologies.

2.1 IoT Overview

This section aims at giving a first description of IoT domain. In particular, we provide a brief description of evolution and history of IoT, and discuss its functional and practical architectures.

2.1.1 IoT Evolution

Nowadays, networks can connect a huge variety of devices. From the most common ones, such as computers or smartphones, to devices like printers, video projectors, etc. The connectivity can be done through cables or wireless signals (such as radio) to allow fast transfer of information. A few decades after its appearance as the interconnection of a massive number of networks, the Internet has been transformed with the help of recent technological advancements into a network where daily used objects can be connected to the network, and recognized and controlled through sensors, smartphones, and so on.

This network of physical objects that can sense, communicate, and be accessed through the Internet is known as the Internet of Things (IoT), which includes devices embedded with electronics, software, sensors, actuators, and network connectivity. The IoT enables the collection and exchange of data using various protocols, offering connectivity that brought several social and industrial sectors into a whole different level of performance, in addition to the emergence of new sectors of activity. For illustration, IoT Analytics[1] states that the number of connected devices worldwide has known an increasing by 1000% (see Figure 2.1) between 2010 and 2020.

1: iot-analytics.com

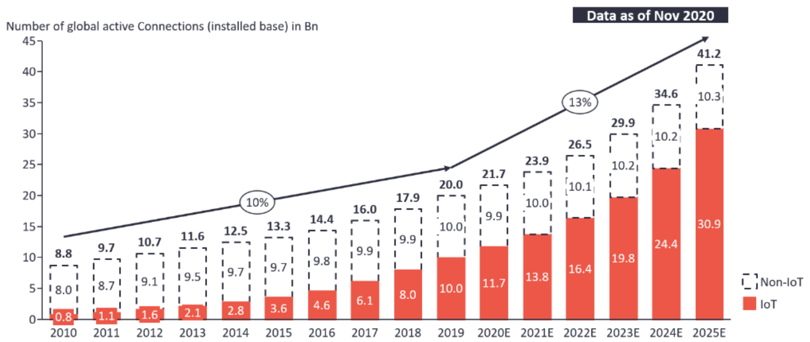


Figure 2.1: Number of Connected Devices Evolution [24].

2.1.2 History

[25]: Sharma et al. (2019), 'The History, Present and Future with IoT'

According to [25], the IoT origin goes back up to 1969, since the first version of the actual Internet appeared that year. The initial objective of Advanced Research Project Agency Network (ARPANET), conducted by Defense Advanced Research Projects Agency (DARPA) was to share research work and to link computers to many general-purpose computing centers of the Defense Department of the United States. A few years after, 1973 was the year of one of the main events in IoT history, through the emergence of an essential technology in IoT, with the first patent of Radio-Frequency Identification (RFID), just before embedded computer systems came into existence in 1974 [26]. Early use of IoT properly speaking and as we know it today (but without being denominated), goes back up to 1984 at Carnegie Melon University, with a coke machine that was connected to the Internet to report the availability and temperature of the drink and even today, these systems are implemented using single board computers and micro-controllers, which are in their turn extremely used in the IoT.

[26]: Manley (1974), 'Embedded Computers: Software Cost Considerations'

The idea of the "Internet of Things" was first introduced with the example of a coffee vending machine in the 1980s, but the term was officially coined by Kevin Ashton, Executive Director of Auto-ID Labs at MIT in 1999 [27]. The concept of IoT gained popularity with the establishment of the Auto-ID Center in 2003 and became a focus of market analysts' publications. From the early days of IoT, various objects were connected to the internet for different applications using a range of technologies, depending on the object and its intended use for the convenience and comfort of humans [28].

[27]: Ashton et al. (2009), 'That 'Internet of Things' thing'

[28]: Madakam et al. (2015), 'Internet of Things (IoT): A Literature review'

Since the 2000s, digitalization has made Internet connectivity a standard for numerous applications, and businesses and products are now expected to have an online presence and provide information online. Despite this and for various reasons, most devices that require human interaction and monitoring through interfaces and applications remain passive entities on the Internet, while most research in IoT is nowadays focused on the miniaturization, the power efficiency and the radio spectrum management of IoT objects and technologies [25].

2.2 IoT Applications

Obviously, the increasing variety of available technologies and devices followed to the appearance of multiple application domains for IoT networks. Some of the notable applications of IoT are described below:

2.2.1 Smart Cities

The use of the IoT domain in the context of Smart Cities has become of interest. The objective of a smart city is to improve the utilization of public resources, particularly in lighting, transportation, and parking, while simultaneously limiting the operational expenses of public administration. Smart city applications can also enhance public safety through surveillance and streamline waste management while providing citizens with better services. Furthermore, data collected from these initiatives can inform citizens about their city's status. However, due to several technical and logistical difficulties, the smart city market has not gained substantial momentum, despite its numerous potential benefits.

According to [29], the major services that can be deployed in a smart city are: (i) Smart metering, (ii) Heating, ventilation and air conditiong (HVAC), (iii) Smart grid and (iv) Environmental monitoring.

[29]: Watteyne et al. (2011), 'Smarter Cities through Standards-based Wireless Sensor Networks'

The usage of IoT in the context of Smart Cities has been extensively studied in the community. [30] analyzes the scalability of Low Power Wide Area Network (LPWAN) for the required mobility in some applications of smart cities (*e.g.,* traffic regulation). In [31], the authors introduce a smart city testing facility located in Antwerp, Belgium, designed to facilitate experimentation at both the technology and user levels in order to address key questions about smart city implementation. The platform offers a multi-wireless technology network infrastructure that enables researchers to easily conduct data experiments and validate their results using a living lab approach. [32] introduces a new communication architecture that is both ubiquitous and resilient. Inspired by the human nervous system, the architecture is designed to be flexible and able to accommodate growth, while still maintaining a high level of performance and reliability. [33] introduces a new security scheme for efficient and secure media packet routing in IoT networks, as well as an algorithm for Media-based Surveillance Systems in IoT networks in the context of Smart Cities.

[30]: Neto et al. (2021), 'Scalability of LPWAN for Smart City Applications'

[31]: Latre et al. (2016), 'City of Things: An Integrated and Multi-technology Testbed for IoT Smart City Experiments'

[32]: Uribe-Pérez et al. (2017), 'A Novel Communication System Approach for a Smart City based on the Human Nervous System'

[33]: Memos et al. (2018), 'An Afficient Algorithm for Media-based Surveillance System (EAMSuS) in IoT Smart City Framework'

2.2.2 Medical and Healthcare

Wireless Sensor Network (WSN) have attracted significant attention due to their important roles in healthcare and medical applications. These networks enable real-time remote monitoring of patients, as well as improving the quality of life for the elderly through smart environments. They also contribute to facilitating drug and medical database management and preventing critical patient situations.

In [34], the authors propose a review of the current state of the integration of remote health monitoring into clinical practice. They state that IoT-enabled wearable sensors provide a promising solution for observing and collecting data in home and work environments over extended periods, surpassing the limited duration of office and laboratory visits. The resulting wealth of data can then be analyzed and presented in intuitive visualizations to doctors. However, there are several obstacles to overcome in terms of sensing, analytics, and visualization before seamless integration into clinical practice can be achieved. [35] focuses on Anomaly Detection in Healthcare and IoT systems. The authors propose a methodology using smartphones for proactive healthcare analytics in preventing cardiac disease. In [36], a proposal was made to use RFID-based applications in body-centric systems to collect information on human behavior while adhering to power and sanitation regulations. [37] proposes an IoT architecture that integrates a Machine Learning (ML) algorithm to detect heart diseases at an early stage. This solution employs a three-tier framework that collects sensor data from wearable devices,

[34]: Hassanalieragh et al. (2015), 'Health Monitoring and Management using Internet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges'

[35]: Ukil et al. (2016), 'IoT Healthcare Analytics: The Importance of Anomaly Detection'

[36]: Amendola et al. (2014), 'RFID Technology for IoT-based Personal Healthcare in Smart Spaces'

[37]: Kumar et al. (2018), 'A Novel Three-tier Internet of Things Architecture with Machine Learning Algorithm for Early Detection of Heart Diseases'

stores the information in the cloud, and employs a regression-based prediction model to identify heart disease.

2.2.3 Smart Agriculture

Smart Agriculture aims at transforming traditional agricultural practices into modern methods adapted to the realities of climate change. The main objective of Smart Agriculture systems is to keep surveillance on factors that can impact crops (soil moisture, temperature change, etc.), through the deployment of different kinds of sensors [38]. These sensors can measure barometric pressure, luminosity, humidity level, etc.

[38]: Cao-Hoang et al. (2017), 'Environment Monitoring System for Agricultural Application based on Wireless Sensor Network'

[39] proposes a framework that combines Remote Monitoring Systems with Internet and wireless communications, providing automatic control of the environmental temperature, and humidity factors, through a friendly interface offering the real-time environmental factors in the greenhouse. [40] presents a crop monitoring system based on a wireless sensor network, where sensors gather data related to temperature and humidity within the testbed. The proposed platform also provides a system for taking periodic crop growth images. In [41], they present an "Intelligent Agriculture Management Information" whose objective is to analyze the features of Agricultural data. The framework is divided into three major steps which are: (i) Data acquisition where crop growth and storage are measured, (ii) data transmission which makes sure the data is transferred to the Internet to be usable by other entities and (iii) information analysis where data mining processes are executed to analyze measured data. [42] provides a review of the potential WSN applications and the specific issues and challenges associated with deploying WSNs for improved farming.

[39]: Zhao et al. (2010), 'The Study and Application of the IoT Technology in Agriculture'

[40]: Liqiang et al. (2011), 'A Crop Monitoring System based on Wireless Sensor Network'

[41]: Yan-e (2011), 'Design of Intelligent Agriculture Management Information System based on IoT'

[42]: Ojha et al. (2015), 'Wireless Sensor Networks for Agriculture: The State-of-the-Art in Practice and Future Challenges'

2.2.4 Smart Home/Building

A Smart Home or Building is characterized by the integration of electronic devices such as heating, lighting, and more. These systems typically consist of sensors and gateways that facilitate communication with a central station. The central station can be controlled through a user interface installed on a mobile phone, tablet, or computer, all of which are managed by IoT technology [43]. In recent years, the concept of Smart Homes/Buildings has evolved, incorporating a range of devices into the IoT.

[43]: Alaa et al. (2017), 'A Review of Smart Home Applications based on Internet of Things'

[44] presents a smart home system including voice activation for switching functions, and a range of devices like light switches, temperature, intrusion detection, smoke/gas sensors, and sirens have been incorporated to demonstrate the practicality and effectiveness of the proposed system. [45] proposes an IoT Smart Home System that allows remote control of household appliances through a mobile device, IR remote control, or PC/Laptop. The system is designed using a WiFi-based microcontroller and includes a temperature sensor to monitor room temperature. [46] describes Frugal Labs IoT Platform (FLIP), which is designed to build Smart Homes that are enabled with IoT technology. Additionally, the paper presents a proposed system that utilizes FLIP to implement Smart

[44]: Kumar (2014), 'Ubiquitous Smart Home System using Android Application'

[45]: Khan et al. (2018), 'Design of an IoT Smart Home System'

[46]: Malche et al. (2017), 'Internet of Things (IoT) for Building Smart Home System'

Home services. The proposed system is primarily used for monitoring and controlling the Smart Home environment.

2.2.5 Smart Manufacturing

The contemporary manufacturing sector is adopting novel technologies such as IoT, big data analytics, cloud computing, and cybersecurity to manage system complexity, enhance information transparency and boost production efficiency. These innovations are facilitating the emergence of a new era of intelligent manufacturing, characterized by a cyber-physical system that closely integrates physical manufacturing enterprises with virtual ones in cyberspace [47].

[47]: Yang et al. (2019), 'The Internet of Things for Smart Manufacturing: A Review'

The authors of [48] have developed a system utilizing IoT to supervise and evaluate energy usage during selective laser sintering[2]. Additionally, a control mechanism was established to enhance efficiency and reduce the overall energy consumption of the process. [49] presents a software application designed for real-time monitoring of energy efficiency on manufacturing shop floors. This software employs data envelopment analysis to identify irregular energy consumption patterns and quantify discrepancies in energy efficiency while monitoring the energy efficiency in real-time. [50] presents an IoT framework that enables the acquisition and integration of real-time data to enhance information visibility across the enterprise, workshop floor, and machine layers. The framework aims to facilitate better decision-making in manufacturing execution.

[48]: Qin et al. (2017), 'A Framework of Energy Consumption Modelling for Additive Manufacturing using Internet of Things'

2: It is a technique that uses a laser as the power and heat source to sinter powdered material to create a solid structure.

[49]: Tan et al. (2017), 'Internet-of-Things enabled Real-time Monitoring of Energy Efficiency on Manufacturing Shop Floors'

[50]: Zhang et al. (2015), 'Real-time Information Capturing and Integration Framework of the Internet of Manufacturing Things'

2.3 IoT Solutions

These different applications and services are rapidly being implemented across various industries and are poised to become commonplace in the near future. To enable and support these services, IoT solutions (or systems) are being developed and deployed. These solutions build upon the Internet protocol stack and inherit the Internet infrastructure's robustness, scalability, and widespread availability. They also possess the potential for network effects, similar to the Internet, where the value and benefits of the IoT ecosystem increase as more devices and services become interconnected. The TENPA (Things, Edge, Network, Platform and Application) model is presented in [51] to represent the end-to-end architecture of an IoT solution. This model is decomposed into five tiers or zones: (i) Things tier, (ii) Edge tier, (iii) Network tier, (iv) Platform tier and (v) Application tier.

[51]: Vicat-blanc (2022), *Systems and Methods for Modeling and Simulating an IoT System*

The **Thing** tier consists of end-devices which are physical objects connected to the Internet and used to collect information from the environment and transmit them. They are equipped with sensors, actuators and other types of hardware to make them able to interact with the physical world. They are also equipped with radio chips to allow them to send/receive information to/from the Internet. To provide intelligence and connectivity to end-devices, the latter are equipped with micro-controllers. Micro-controllers are integrated circuits containing processing units, memory, input/output peripherals, radio units, etc. They can be programmed with firmware to perform specific tasks.

Firmware Operating Systems: Firmware is run in the micro-controllers to control their components. They can be implemented using diverse operating systems. Due to the energy- and space constraints, these operating systems must be minimalist. Some of them are also referred to as Real-Time Operating System (RTOS) [52]. We present in what follows some well-known operating systems for IoT:

[52]: Stankovic et al. (2004), 'Real-time Operating Systems'

[53]: Baccelli et al. (2013), 'RIOT OS: Towards an OS for the Internet of Things'

- **RIOT**: RIOT [53] is an open-source operating system specifically designed for IoT. It takes into account the limitations of devices with minimal resources while facilitating development across a wide range of IoT devices. Its design objectives include energy efficiency, reliability, real-time capabilities, a small memory footprint, and modularity, independent of the underlying hardware.

[54]: Dunkels et al. (2004), 'Contiki-a Lightweight and Flexible Operating System for Tiny Networked Sensors'

- **Contiki**: Contiki [54] is an open-source operating system designed for IoT devices. Its focus is on reliable and secure low-power communication. This operating system is accompanied by comprehensive documentation and is widely used in the IoT community. Note that Contiki is not, stricto sensu, an RTOS. Indeed, The Contiki programming model is based on protothreads [55], which are lightweight threads providing a blocking context on top of an event-driven system, while typical RTOSs employ preemptive scheduling, where higher-priority tasks can interrupt lower-priority ones.

[55]: Dunkels et al. (2005), 'Using Protothreads for Sensor Node Programming'

[56]: Barry et al. (2008), 'FreeRTOS'

- **FreeRTOS**: FreeRTOS [56] is a minimalistic and straightforward operating system designed to be compact in size. It is primarily coded in C programming language to simplify its portability and maintenance. However, some assembly language functions are also incorporated, primarily in architecture-specific scheduler routines.

[57]: Zephyr (2005), *Zephyr Project Documentation*

- **Zephyr**: Zephyr [57] is an RTOS dedicated to connected, resource-constrained and embedded devices (with an emphasis on micro-controllers), supporting multiple architectures. It contains a kernel, and all components and libraries, device drivers, protocol stacks and firmware updates, needed to develop full application software.

The **Edge** tier refers to the interface between physical devices and the wider IoT network. It encompasses the functionalities of protocol adaptation, data filtering, data collection, pre-processing, and providing internet connectivity to enhance the capabilities of IoT devices.

The **Network** tier is composed of gateways that are devices serving as bridges between end-devices and the Cloud. It is responsible for facilitating the transportation of IoT data over long distances and through Internet connections to centralized data centers or Cloud platforms.

The **Platform** tier serves as a hub for connection aggregation, security functions, and data management tasks such as data exchange, processing, and storage.

The **Application** tier is composed of applications/services and cloud servers which are designed and implemented principally for data storage and further functionalities, such as web monitoring and mobile applications. It can be the entry point of an end-user to monitor the state of the network or to execute commands in the end-devices.

Still according to [51], the Things and Application tiers are mandatory. The network tier makes this system an Internet-based system, while the

Edge and Platform tiers' role is to enhance the overall performance and usability of the IoT system.

The way to arrange end-devices and gateways is known as the topology. We distinguish two major topologies in IoT:

Star: Every node (end-device) is attached to a node that is central in a star topology and is connected to the Internet. The central node is typically called the gateway. It is responsible for receiving (respectively sending) data from sensors (respectively to actuators), as well as processing data. Figure 2.2 displays a star topology, where the central node acts as a gateway and all other nodes attached to this node are the end-devices.

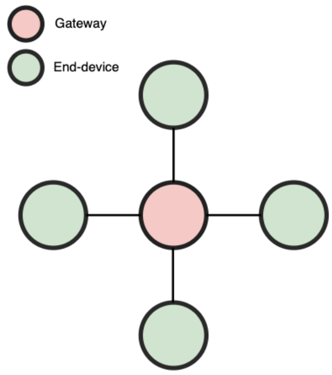


Figure 2.2: Star Topology.

Mesh: It allows nodes to connect to many other nodes so that the Internet gateway can be one or more nodes. Figure 2.3 depicts a mesh topology where all nodes are interconnected. Although simple in real life, designing a mesh network can be challenging and may result in longer intervals for messages to travel between distant nodes in comparison to a star topology. There are also routing issues that may appear when using a mesh topology.

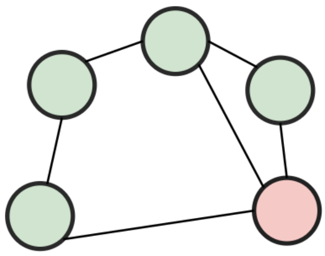


Figure 2.3: Mesh Topology.

2.4 IoT Network Technologies

As mentioned in the previous section, the goal of IoT network technologies is to establish a connection between the physical world (the "things") and the virtual world (the applications or end-user services). An IoT network technology is a wireless technology typically defined by the two lower layers of the Open Systems Interconnection (OSI) model [58] (Physical and Data Link). They are by definition dependent on the communication medium used, and must therefore take into account the specificities of the latter. Since the communication medium is air which is shared by all nodes, wireless technologies have important differences from wired technologies (which benefit from isolated communication in independent cables).

[58]: Zimmermann (1980), 'OSI Reference Model-the ISO Model of Architecture for Open Systems Interconnection'

IoT network technologies are characterized by features, which are inherent characteristics, regardless of the targeted application. Some of them are presented in the following:

- **Data rate**: It is the theoretical maximal amount of data that can be sent per unit of time.
- **Range**: Range refers to the theoretical maximal distance that a packet can cross over the air with enough power to be decoded by the receiver. This value can be affected by the radio conditions and environment. For instance, packets can typically traverse longer distances in a rural environment than in an urban one, due to the fewer interferences and obstacles.
- **Security**: Since the IoT is an open ecosystem in which all devices are interconnected, these devices are vulnerable to malicious attacks. Due to this risk, several research efforts have focused on mechanisms to achieve reliable, at the same time, lightweight IoT security and privacy. The robustness of the used encryption algorithm to cipher the transmitted data is often used to determine the security of an IoT network technology [59–61].

- **Cost**: Cost is a complex aspect to model in IoT. It generally englobes the fees induced by deployment (spectrum cost in case of licensed bands, price of end-devices and gateways, etc.) and maintenance (battery replacements, etc.). We can see that although some aspects are inherent in the network technology, it is greatly dependent on the application.

Some technologies provide coverage in tens of meters, and they are named short-range technologies, while others can provide coverage up to tens of kilometers, and they are called long-range technologies. A description of both types of network technologies is provided in the following:

2.4.1 Short-Range Technologies

Wireless Personal Area Network (WPAN) provide coverage that does not exceed hundreds of meters. They are designed for relatively proximity communication between devices. These technologies are typically used for IoT applications that operate within a limited range, such as within a room or a building. Additionally, short-range technologies often allow higher data rates than long-range ones. Some common short-range technologies used in IoT include:

- ***Wi-Fi***
  Wi-Fi is a WLAN technology that is based on the IEEE 802.11 standard. It is widely used. As of 2019, over 3.05 billion Wi-Fi-enabled devices are shipped globally each year [62].
  Wi-Fi can operate on several frequency bands: 900 MHz, 2.4 GHz, 3.6 GHz, 4.9 GHz, 5 GHz, 5.9 GHz, 6 GHz and 60 GHz. It uses the listen-before-talk mechanism Carrier Sense Multiple Access (CSMA) which checks before sending a frame that the medium is idle. Wi-Fi stations and their access points (gateways) typically have a wireless communication range of about 30 meters indoors. Wi-Fi data rate depends on the amendment. For example, for 802.11a, b, g, n, ac and ax, data rates can be up to 54 Mbps, 11, 54, 150, 866.7 Mbps and 7 Gbps, respectively [63]. Several parameters affect Wi-Fi connection, like the Modulation and Coding Scheme (MCS). The latter refers to the combination of modulation and coding to achieve different data rates and levels of robustness in the presence of noise and interference. Typically, higher MCS will lead to a higher data rate, but also a higher sensitivity from the transmission to noise and interference.
  Wi-Fi generally has a star topology, even though it also allows Point to Point (P2P) communications (ad-hoc) between end-devices [64]. Energy consumption was not a real concern when Wi-Fi was developed and it was not initially developed for IoT purposes, since it was designed to provide broadband wireless internet access for a small number of plugged-in stations. However, an amendment to the Wi-Fi standard IEEE 802.11ah (also known as Wi-Fi HaLow) has been developed to make Wi-Fi more adapted for IoT, by increasing the coverage of access points up to 1 km and, making them able to handle up to 8,000 stations. To date, 802.11ah is the only Wi-Fi version that supports low-energy communications of a high number

[62]: Market Intelligence & Consulting Institute (MIC) (2020), 'Global Wi-Fi Enabled Devices Shipment Forecast, 2020-2024'

[63]: Parmar et al. (2016), 'IoT: Networking Technologies and Research Challenges'

[64]: Noreen et al. (2017), 'A Study of LoRa Low Power and Wide Area Network Technology'

[65]: Khorov et al. (2019), 'Enabling the Internet of Things with Wi-Fi Halow—Performance Evaluation of the Restricted Access Window'

of IoT stations placed in a large area [65]. Despite its introduction to the community in 2017, Wi-Fi HaLow has not been widely adopted, and there are currently only a limited number of implementations available. Further details about Wi-Fi can be found in [66].

[66]: Gast (2005), *802.11 Wireless Networks: The Definitive Guide*

- ***BLE***
  Bluetooth Low Energy (BLE) is a low-power wireless technology developed for short-range control and monitoring applications. In 2020, the market volume of BLE-enabled devices has been estimated to be 8 billion units*. BLE operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) band and defines 40 Radio Frequency (RF) channels with 2 MHz channel spacing. All physical channels use a Gaussian Frequency Shift Keying (GFSK) modulation.
  Bluetooth-based networks can be deployed in a P2P or in a star topology [64]. It uses a Time Division Multiple Access (TDMA) schemes, where the central station determines the instants in which other stations are required to listen, and thus coordinates the medium access [67]. It can reach data rates up to 1 Mbps, and a range of up to 100 meters [64].
  In contrast with the previous Bluetooth version, BLE has been designed as a low-power solution for control and monitoring applications.

[67]: Gomez et al. (2012), 'Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-power Wireless Technology'

- ***802.15.4***
  The IEEE 802.15.4 [68] is a technical standard that defines the physical and mac layers of low-rate Low Rate Wireless Personal Area Network (LR-WPAN). It can operate on the 868 MHz, 915 MHz and 2.4 GHz frequency bands, with Direct Sequence Spread Spectrum (DSSS) modulation. It provides wireless data transmission, intending to let stations communicate over small ranges [60].
  ZigBee is a short-range technology based on IEEE 802.15.4. An estimation stated that by 2023, there will be 4.5 billion 802.15.4 mesh devices sold worldwide, most of which will use Zigbee†. It supports star and mesh topologies [64]. Like Wi-Fi, it uses the CSMA mechanism for controlling the medium access. Its data rate can go up to 250 Kbps and its range varies between 10 and 100 meters [64].
  The IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [69] is a communication protocol designed to enable the transmission of IPv6 packets over LR-WPAN networks. It is based on the IEEE 802.15.4 norm, and it defines the mechanisms for compressing IPv6 packets, fragmenting and reassembling them to fit within the limited payload size of 802.15.4 frames and handling the addressing and routing of packets in a 6LoWPAN network.
  The IPv6 over the Time Slotted Channel Hopping[3] (TSCH) mode of IEEE 802.15.4e (6TiSCH)[70] proposes a protocol stack rooted in the mode of the IEEE 802.15.4, supports multi-hop topologies with the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) routing protocol, and is IPv6-ready through 6LoWPAN.

[68]: IEEE Computer Society (2006), 'Part 15 . 4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)'

[60]: Ismaili et al. (2019), 'Comparative Study of ZigBee and 6LoWPAN Protocols'

[64]: Noreen et al. (2017), 'A Study of LoRa Low Power and Wide Area Network Technology'

[69]: Mulligan (2007), 'The 6LoWPAN Architecture'

3: It is a channel access method where time is divided into slots and each slot is assigned a specific frequency channel for transmission.

[70]: Vilajosana et al. (2019), 'IETF 6TiSCH: A Tutorial'

- ***RFID***

* https://statista.com/statistics/750569/worldwide-bluetooth-low-energy-device-market-volume/

† https://zigbeealliance.org/news_and_articles/zigbee-leads-the-wireless-mesh-sensor-network-market/

Radio-Frequency Identification (RFID) is a wireless system comprised of two components: tags and readers. The reader is a device that has one or more antennas that emit radio waves and receive signals back from the RFID tag. It has two different versions. Low Frequency (LF) RFID operates at a frequency of 125-134 kHz which provides a short read range of 10 cm. So-called Long-range RFID, also known as RAIN RFID, operates on the Ultra High Frequency (UHF) band and offers fast recognition speed, with the ability to read tags up to 15 meters. On the other hand, so-called low-range RFID uses the High Frequency (HF) band, which has a lower tag recognition range (up to 1.2 meters) [71]. Finally, Microwave Frequency (MW) RFID operates at a frequency range of 2.45-5.8 GHz and has a range of under 2 meters [72].

[71]: Elbasani et al. (2020), 'A Survey on RFID in Industry 4.0'

[72]: Pillai et al. (2007), 'An Ultra-low-power Long Range Battery/passive RFID Tag for UHF and Microwave Bands with a Current Consumption of 700 nA at 1.5 V'

2.4.2 Long-Range Technologies

Long-Range network technologies offer radio coverage over a large area by way of base stations (gateways). [73] classifies them according to the type of frequency bands they operate on: (i) ISM band-based and (ii) Mobile band-based. Both are described in the following:

[73]: Foubert et al. (2020), 'Long-range Wireless Radio Technologies: A Survey'

ISM band-based

Contrary to mobile IoT network technologies, the ISM-based LPWAN technologies use unlicensed bands in the spectrum. These bands can be used by anyone without having to possess an authorization, which naturally augments the contention on the medium.

- ***LoRaWAN***
  Over 1.2 million gateways and 180 million devices using LoRa are deployed all around the world today‡. It utilizes a combination of Chirp Spread Spectrum (CSS) modulation at the physical layer [74] and LoRaWAN at the MAC layer. CSS modulation is based on frequency ramps with cyclic shifts, which can encode information using a variable Spreading Factor (SF). The latter represents the amount of spreading code applied to the original data signal. LoRa modulation has a total of six spreading factors (SF7 to SF12). The larger the used spreading factor, the farther the signal will be able to travel and still be received without errors by the receiver. This parameter affects the communication range, the time that a packet takes to arrive and the data rate. This modulation scheme is highly resilient against interference and the Doppler effect [4], allowing it to achieve long-range transmissions.
  LoRa technology operates in unlicensed ISM bands, initially in the EU863-870 MHz and the EU433 MHz Sub-GHz bands in Europe, as well as in some countries in Africa and the Middle East. Recently, it has expanded its operations to include the 2.4 GHz band. LoRaWAN does not use any listen-before-talk mechanism, since it uses pure ALOHA access method. It enables low power operations (around 10 years of battery lifetime), offering low data

[74]: Berni et al. (1973), 'On the Utility of Chirp Modulation for Digital Signaling'

4: It refers to the change in wave frequency during the relative motion between a wave source and its observer.

‡ https://www.semtech.com/lora

rate and long communication range (2-5 km in urban areas and 15 km in suburban areas) [75].
Although LoRaWAN does not use any listen-before-talk mechanism before sending packets, it undergoes restriction and a key limiting factor, which is the duty-cycle regulations in the ISM bands: In Europe, the maximum duty-cycle of the EU 868 ISM band is 1%, which means that if a packet takes a time t to arrive, the sending device must observe a waiting time equal to $100 * t$ before sending another packet.
LoRaWAN has a star topology network composed of end-devices and gateways connected through the Internet to a network server. There are three classes of end-devices in LoRaWAN: (i) Class A, where devices can send a packet and receive only in an interval after its emission, (ii) Class B which is the same as class A but devices of class B can receive at regular intervals, (iii) Class C where devices can continuously receive message. Naturally, Class A is the class of LoRaWAN devices with the lowest power consumption.

[75]: Adelantado et al. (2017), 'Understanding the Limits of LoRaWAN'

- ***Sigfox***
  Sigfox is an LPWAN network operator that offers an end-to-end IoT connectivity solution based on its patented technologies. It uses a Binary Phase Shift Keying (BPSK) modulation in an Ultra Narrow Band (UNB) of 100 Hz.
  Just like LoRaWAN, Sigfox operates on the EU863-870 MHz and the EU433 MHz Sub-GHz bands in Europe. The used mechanism for accessing the medium is also ALOHA. By employing the ultra-narrow band, Sigfox can ensure very low power consumption, high receiver sensitivity, and low-cost antenna design at the expense of a maximum throughput of only 100 bps. Its range can however go up to 10 km (urban) and 40 km (rural) [59].
  In the uplink direction, the packet size is restricted to a maximum of 12 bytes. Additionally, Sigfox applies the 1% duty cycle for ISM bands in Europe. To improve reliability, the system employs redundant transmissions and time-frequency diversity, which involves transmitting multiple times over randomly selected channels. However, acknowledgments are not supported by the system.
  Sigfox is also characterized as a network with a star topology.

[59]: Mekki et al. (2018), 'Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT'

Figure 2.4§ displays this classification, showing the trade-off between the range and the data rate. Note that cellular networks are the only exception to this trade-off. Also note that the cost reflects a global overview of the cost of using a network technology (including the licensed bands and equipment price, etc.).

Mobile band-based

Mobile technologies are extremely used over the world nowadays. They operate on a licensed spectrum. Due to their impressive coverage, they may well be the widest wireless networks deployed to provide Internet access. A recent interest of the 3rd Generation Partnership Project (3GPP) consortium, is to allow the exploitation of IoT applications using mobile

§ https://embeddedams.nl/

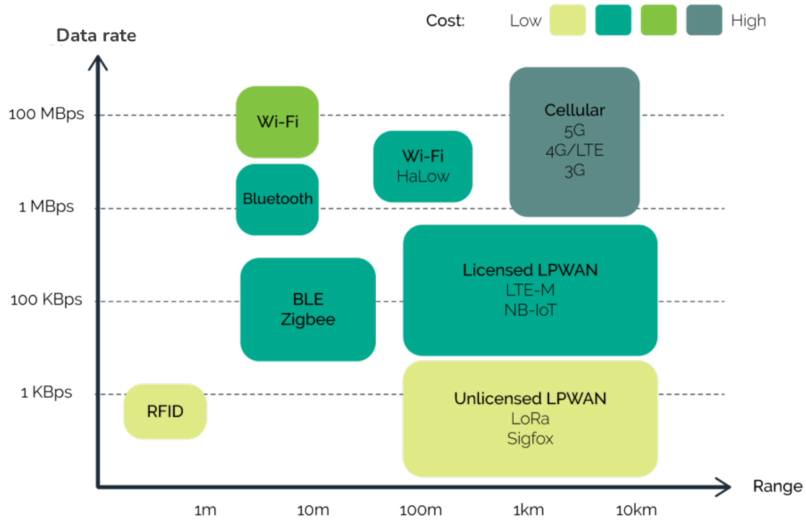


Figure 2.4: IoT Network Technologies Classification (https://embeddedams.nl/).

communications, with respect to IoT special constraints, mainly in terms of computation power and energy restriction. We present in what follows some well-known mobile network technologies specially designed for the IoT:

- ***NB-IoT***
  Narrowband-IoT (NB-IoT) has been defined by 3GPP based on the characteristics of Long Term Evolution (LTE) to enable its rapid adoption and seamless integration into existing LTE networks, with simple modifications considering the use cases of IoT, particularly related to extended coverage and low power consumption. NB-IoT incorporates a substantial portion of the LTE design. This includes the use of downlink Orthogonal Frequency-Division Multiple Access (OFDMA), uplink Single-Carrier Frequency-Division Multiple Access (SC-FDMA), channel coding, rate matching, and interleaving [76]. It uses Gaussian Minimum Shift Keying (GMSK) for the modulation.
  Unlike LoRaWAN and Sigfox, NB-IoT uses the SC-FDMA mechanism, where multiple access among users is made possible by assigning to different users different sets of non-overlapping subcarriers. NB-IoT range can go up to 1 km (urban) and 10 km (rural) and can ensure data rates up to 200 kbps [59].
  NB-IoT also supports star topology. NB-IoT is kept as simple as possible to reduce device costs and minimize battery consumption, and thus it removes many features of LTE, including handover, measurements to monitor the channel quality, carrier aggregation, and dual connectivity [77].

[76]: Schlienz et al. (2016), 'Narrowband Internet of Things Whitepaper'

[59]: Mekki et al. (2018), 'Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT'

[77]: Sinha et al. (2017), 'A Survey on LPWA Technology: LoRa and NB-IoT'

- ***LTE-M***
  LTE-M is a machine-focused variant of the 3GPP LTE standard, which is designed to meet the high-coverage, low-cost, and low-power consumption requirements of the IoT. It is already deployed in numerous countries worldwide. It also uses the OFDMA mechanism in the MAC layer. Its range can go up to 11 km and its data rate can go up to 1 Mbps [73]. It uses the Quadrature Phase Shift Keying (QPSK) modulation scheme.

- ***5G***
  The widespread of the fifth generation of mobile networks (5G) began in 2019, and 5G networks are predicted to have more than 1.7 billion subscribers worldwide by 2025¶. 5G utilizes multi-layer spectrum through the use of large-scale antennas, with sub 1 GHz for low-band spectrum, 1 GHz and 6 GHz for mid-band spectrum, and 24-40 GHz for high-band spectrum Millimeter Wave (mmWave). It uses the Orthogonal Frequency-Division Multiplexing (OFDM) for the modulation.
  5G uses the OFDMA mechanisms for medium access by assigning subsets of subcarriers to individual users. With 5G, up to 1 million devices can be connected per square km, with the ability to communicate at speeds of up to 500 km/h. This technology also supports uplink speeds of at least 10 Gbps and downlink speeds of up to 20 Gbps, with download speeds of 100 Mbps and upload speeds of 50 Mbps per user. The latency resulting from the use of 5G technology should be around 1 ms [78].

  [78]: Andrews et al. (2014), 'What will 5G be?'

  5G can support star and mesh topologies. Although mobile technologies have recently been a great part of our lives, previous mobile generations like 3G or 4G were not considered facilitators for IoT, unlike 5G. Indeed, a huge improvement in network scalability, connectivity and energy efficiency, and a range of about a couple of kilometers, have made that technology a key enabler for many IoT services [79].

  [79]: Mitra et al. (2015), '5G Mobile Technology: A Survey'

- ***6G***
  Even though 5G has brought wireless systems to a different level in terms of QoS, it will presumably be insufficient for supporting the unprecedented increase in the number of connected devices and traffic volume demand [80]. In this context, 6G, which is the sixth generation of wireless technology, is expected to introduce innovative wireless technologies and networking infrastructures that can meet the demanding requirements of a wide range of new IoT applications. Moreover, it is expected that it satisfies these requirements in a more comprehensive manner than the current 5G technology, which is expected to lead to significant disruptions in the IoT ecosystem [81].

  [80]: Chowdhury et al. (2020), '6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions'

  [81]: Nguyen et al. (2021), '6G Internet of Things: A Comprehensive Survey'

  Although it is still in the research and development phase, 6G is supposed to allow faster communications, lower latency, higher capacity, and improved reliability compared to 5G. Indeed, 6G shall provide data rates up to 1 Tbps, latency under 1 ms and will be able to support up to 10 million devices per km^2 [82, 83]. It is also likely to enable new use-cases and applications such as advanced virtual and augmented reality, and more advanced automation and robotics. Its deployment is planned to be starting in 2030.

  [82]: Yap et al. (2022), 'Future Outlook on 6G Technology for Renewable Energy Sources (RES)'
  [83]: Dogra et al. (2020), 'A Survey on Beyond 5G Network with the Advent of 6G: Architecture and Emerging Technologies'

  6G is envisioned to revolutionize customer services and applications via IoT, towards a future of fully intelligent and autonomous systems. Therefore, one of the major goals of 6G is to build less energy-intensive systems. Indeed, typical wireless cellular network

¶ https://www.forest-interactive.com/insights/5g-connections-post-covid-19-global-forecast/

[84]: Al-Turjman et al. (2017), 'Energy Efficiency Perspectives of Femtocells in Internet of Things: Recent Advances and Challenges'

base stations consume a considerable amount of power [84]. This indicates that the deployment of large-scale 6G-IoT networks comprising thousands of such stations can result in a significant amount of energy consumption, leading to an increase in carbon emissions. Therefore, in the current context of climate change, developing energy-efficient communication protocols through optimization techniques is an absolute necessity to establish eco-friendly 6G-IoT networks. For that sake, a huge research effort is made toward the development of less consuming mechanisms for 6G [85–87]. Moreover, energy harvesting techniques to exploit renewable energy resources will be very useful to build green 6G-IoT systems [88].

[85]: Zhen et al. (2020), 'Energy-efficient Random Access for LEO Satellite-assisted 6G Internet of Remote Things'

[86]: Mukherjee et al. (2020), 'Energy-efficient Resource Allocation Strategy in Massive IoT for Industrial 6G Applications'

[87]: Sodhro et al. (2020), 'Toward 6G Architecture for Energy-efficient Communication in IoT-enabled Smart Automation Systems'

[88]: Zhao et al. (2017), 'Exploiting Interference for Energy Harvesting: A Survey, Research Issues, and Challenges'

Others

Several proprietary network technologies have emerged in the industry. Ingenu [89] is a proprietary LPWAN technology that uses the 2.4 GHz ISM band (which is already being used by other technologies such as Wi-Fi) instead of sub-GHz ISM bands. The advantage of using this band is that it is not subject to heavy-duty cycle restrictions as the sub-GHz bands are. Ingenu uses a proprietary physical technology called Random Phase Multiple Access (RPMA), which is a variation of Code Division Multiple Access (CDMA). Ingenu states that their technology outperforms most other LPWAN technologies, particularly in terms of range and data rate, with an up-link rate of 78 kbps and a downlink rate of 19.5 kbps [89]. However, [73] states that there is no scientific study available to validate this claim. Weightless [90] is a set of standards (W, P and N) developed by the Weightless Special Interest Group (Weightless-SIG). Weightless-P uses sub-GHz frequencies and offers a data rate of up to 100 kbps with a range of up to 10 km. Weightless-N is a narrow-band IoT technology that uses sub-GHz frequencies and offers a data rate of up to 100 kbps with a range of up to 2 km. Weightless-N is based on Ultra Narrow Band (UNB) for upward-only communications, using the ISM sub-GHz bands. Weightless-P is the latest Weightless standard. It offers bidirectional connectivity based on the ISM sub-GHz bands as well, using channels 12.5 kHz wide which results in a data rate between 0.2 kbps and 100 kbps. GMSK and QPSK modulations are used [73]. The DASH7 Alliance [91] is an industry consortium that proposes a complete stack protocol called DASH7 Alliance Protocol (D7AP), which is based on narrow-band modulation in the ISM sub-GHz bands. D7AP comprises a complex network stack and includes features like periodic wake-up of the nodes, leading to reduced latency in communication, but also increasing power consumption. The protocol offers a data rate ranging from 9.6 to 166.7 kbps [73]. DECT NR+ is a standard developed by European Telecommunications Standards Institute (ETSI), operating in the 1.9 GHz band, designed to create a decentralized mesh network independent of wireless operators. It offers high data rates (from 3 to 80 Mbps), and is the first non-cellular 5G standard, particularly suited for Massive Machine Type Communications (mMTC). DECT NR+ employs OFDM with Turbo channel coding and adaptive modulation, allowing the creation of private wireless networks without the need for existing cellular infrastructure. Recent analyses have shown DECT 2020 NR software, when applied to Bluetooth silicon, to be 24 times more efficient than Narrowband Internet

[73]: Foubert et al. (2020), 'Long-range Wireless Radio Technologies: A Survey'

Table 2.1: IoT Technologies Features.

Technology \ Feature	Data Rate	Range	Security (Encryption)	Cost (Spectrum)
Wi-Fi	[54 Mbps - 7 Gbps]	100 m	WPA-TKIP	Unlicensed
BLE	1 Mbps	50 m	SAFER+	Unlicensed
Zigbee	250 Kbps	100 m	AES 128 b	Unlicensed
LoRaWAN	[3 Kpbs - 50 Kbps]	Urban: 5 km Rural: 20 km	AES 128 b	Unlicensed
Sigfox	Uplink: 100 bps Donwlink: 600 bps	Urban: 10 km Rural: 40 km	No encryption	Unlicensed
NB-IoT	Uplink: 158.5 Kbps Donwlink: 106 Kbps	Urban: 1 km Rural: 10 km	LTE encryption	Licensed
5G	Uplink: 10 Gbps Downlink: 20 Gbps	Low bands: 10 km High bands: 1 km	128-bit encryption	Licensed

of Things (NB-IoT), and when applied to cellular chips, it exhibited a 2.4 times improvement in power efficiency over NB-IoT devices [92].

[92]: Shay (2020), 'DECT 2020 NR : Sustainability and Scale for the IoT (White Paper)'

Besides that, considering that IoT is supposed, in some cases, to cover a large geographical area or a region not reached by terrestrial network connection, the usage of satellites for IoT might be of great interest for extending the coverage and the connectivity in a flexible and affordable manner. According to [93], the revenues generated by the global satellite IoT connectivity industry will more than double from 233 million USD in 2019 to 554 million USD in 2025. Furthermore, the number of cumulative satellite connections is expected to increase fourfold, reaching over 10 million by the year 2025. In this perspective, traditional communication service providers and vendors spare no expense on exploiting the existing terrestrial wireless long-range technologies as ground-to-satellite link enabler [94]. The architecture of a Satellite IoT system comprises the following components:

[93]: Lucero (2020), 'Satellite IoT Market Report 2020'

[94]: Centenaro et al. (2021), 'A Survey on Technologies, Standards and Open Challenges in Satellite IoT'

- The end-devices, also referred to as Machine Type Devices (MTD)s diposed in the terrestrial environment,
- IoT-support satellites, which can be seen as gateways for satellites,
- IoT gateways (for the terrestrial MTD),
- Ground stations, which gather and transfer data from IoT-support satellites to MTD,
- IoT services, providing core network functionalities to perform as well as external cloud applications for user-plane data management.

[95]: Ahad et al. (2020), 'Technologies Trend towards 5G Network for Smart Health-Care using IoT: A Review'

Table 2.1 (inspired from [95]) gives a summary about some IoT technologies and their respective features.

2.5 Conclusion

Our aim through this chapter was to provide a global overview of the different concepts of IoT used in the remainder of this thesis. After a brief description of the history of IoT and why is it that it became so omnipresent in our daily lives, we presented some of the major IoT applications that are witnessing active interest from the community as well as some recent research works that have been conducted in the context of each application. Then, we described the architecture of IoT solutions, before moving to the description of some of the most employed network technologies in the industry and precising for each one of them some of its characteristics in terms of range and data rate as well as the mechanisms it uses on the Physical (PHY) and MAC layers.

State of the Art 3

We present in this chapter an overview of the different approaches for evaluating, selecting and configuring IoT network technologies for a given application. First, we present the classical field approaches that IoT architects usually rely on. Then, we present the scientific approaches that have been proposed in the literature by the research community.

3.1 Field Approaches

Before implementing a full-scale IoT solution, IoT architects usually refer to two field approaches: (i) Proof of Concept (PoC) and (ii) pilot projects. PoC test consists of designing and executing small-scale tests to validate the feasibility and effectiveness of the solution in real-world scenarios. These tests help assess the solution's performance, reliability, and ability to meet the organization's specific requirements. IoT architects analyze the results of PoC tests and provide recommendations based on their findings.

Pilot projects involve deploying and testing IoT solutions in real-world environments on a larger scale than PoC. They allow for practical evaluation of the solution's performance, functionality, and usability. Pilot projects provide valuable insights into how the solution operates in specific contexts and help identify any challenges or areas for improvement. Typically, pilot projects allow the evaluation and calibration of various hardware and physical parameters to achieve optimal adjustments. However, they provide only a limited and incomplete representation of the future reality, as they usually involve a small number of devices and a restricted scope.

Although PoC and pilot projects can be useful tools for IoT architects, a broader and long-term perspective during the design phase would be of interest. Unfortunately, conducting large-scale and long-term experiments is complex due to cost, time, and complexity. Moreover, the constantly evolving landscape of IoT network technologies and architectures, such as 5G, satellite, and edge settings, further adds to the complexity and confusion. Undeniably, testing sensing and network technologies through real-world deployments is essential for assessing functional feasibility. It is crucial to verify the compatibility of sensors with the chosen network technology and validate the quality of collected and transmitted data. Understanding the impact of the connected solution on existing human processes is also vital.

When the choice for the most adequate network technology must be made for a forthcoming IoT solution, IoT architects also refer to IoT surveys. Indeed, they can be a valuable source of information regarding the modulations in use, the channel bandwidth, the maximum payload sizes, and the authentication and encryption support. Surveys cover IoT communication in general (*e.g.*, [10, 96]), or are more specifically devoted to LPWAN technologies (*e.g.*, [59, 77, 97, 98]). Either way, these

[10]: Kassab et al. (2020), 'A–Z survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations'
[96]: Kaňuch et al. (2020), 'Survey: Classification of the IoT Technologies for Better Selection to Real Use'
[59]: Mekki et al. (2018), 'Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT'
[77]: Sinha et al. (2017), 'A Survey on LPWA Technology: LoRa and NB-IoT'
[97]: Ikpehai et al. (2018), 'Low-power Wide Area Network Technologies for Internet-of-Things: A Comparative Review'
[98]: K. Mekki, E. Bajic, F. Chaxel, and F. Meyer (2019), 'A Comparative Study of LPWAN Technologies for Large-scale IoT Deployment'

surveys have limitations: They provide coarse-grained information for networking performance metrics, often limited to best and worst-case values, independent of the targeted IoT applications. Yet, in IoT, the choice of the right network technology is strongly tied to the specific requirements of the application, the network topology, the environment as well as the resources embedded within end-devices. For instance, the theoretical capacity of network technologies can be misleading as in practice, this value will often not be reached because of (i) the channel errors and/or interference, and (ii) the contention resulting from end-devices attempting to access the radio channel at the same time. Also, surveys can be rapidly outdated (there is a constant need for a fresh and up-to-date view in such a prolific and dynamic domain). More importantly, their technical description is often too general and impractical to IoT architects looking for a precise answer to the specific needs of their application.

3.2 Scientific Approaches

In contrast with the field approaches, scientific approaches for IoT solutions design decision-making have emerged. First, we present the evaluation tools that the scientific community relies on for the rigorous evaluation of network technologies. Then, we present a state-of-the-art of scientific methods that have been proposed in the literature.

3.2.1 IoT Network Evaluation Tools

Different evaluation tools have been introduced and can be used for assessing the performance of a given IoT network technology. The following are presented as the principal tools for the evaluation of IoT network technologies.

Testbeds

Testbeds are platforms used for conducting experiments and testing protocols, solutions, etc. Some well-known testbeds in IoT are presented in what follows:

[99]: Adjih et al. (2015), 'FIT IoT-LAB: A Large Scale Open Eperimental IoT Testbed'

FIT IoT-LAB FIT IoT-Lab [99] is a large-scale open-access testbed located in France across 6 different sites (Grenoble, Strasbourg, Lille, Paris, Saclay and Lyon). It provides access to more than 2,300 sensor nodes available to users for experiments in embedded wireless communications. It offers different experimentation boards, different radio technologies, namely 802.15.4 and LoRa, and multi-Operating System (OS).

[100]: Sanchez et al. (2014), 'SmartSantander: IoT Experimentation over a Smart City Testbed'

Smart Santander SmartSantander [100] is a city-scale testbed mostly dedicated to experimenting IoT services targeted at smart cities. It is operating in 4 different countries (Spain, Serbia, Germany and the United Kingdom). It includes a high range of heterogeneous sets of sensors comprising 802.15.4 sensor nodes, parking sensors, RFID, etc.

[101]: Bouckaert et al. (2011), 'The w-iLab.t Testbed'

w-iLab.t w-iLab.t [101] is an open-source testbed deployed in Ghent, Belgium. It supports large-scale sensor deployments, Wi-Fi-based mesh and

Table 3.1: Comparative Table of IoT Testbeds.

Testbed	Scale	Environment type	Node Heterogeneity	Mobility	Built-in IoT standards	Target domain
FIT IoT-LAB [99]	Medium (>2,700 sensors)	Laboratory-like environment	Yes	Yes (robot-driven)	802.15.4 LoRaWAN BLE UWB	Protocol and algorithm performance analysis
Smart Santander [100]	Large (≈ 20,000 sensors)	Real city	Yes	Yes (vehicle-based)	802.15.4 RFID	Smart city IoT service development
https://my.visualstudio.com/ProductKeys w-iLab.t [101]	Small (< 100 sensors)	Laboratory-like environment	Yes	Yes	802.15.4 802.15.4g LTE 802.11a/b/g/n/ac Bluetooth	Protocol and algorithm performance analysis

ad hoc tests, and mixed sensor/Wi-Fi experiments, and is, therefore, able to analyze the behavior of future heterogeneous network deployments.

Table 3.1 compares the presented IoT testbed (inspired from [102]). The scale refers to the number of end-devices deployed in the testbed, while the environment type defines the practical applicability. The ability to support different types of nodes and protocols is defined by Heterogeneity. Mobility indicates the support of mobile end-devices. Concurrency is the ability to support multiple distinct parallel evaluations at the same time. The list of supported IoT network technologies is given in the Built-in IoT standards. Finally, the Target domain is the primary use-case targeted by the testbed.

[102]: Chernyshev et al. (2017), 'Internet of Things (IoT): Research, Simulators, and Testbeds'

Several other testbeds that support fewer network technologies have been developed for IoT purposes. The JOSE [103] testbed, situated in Japan, utilizes the Infrastructure as a Service (IaaS) model for evaluating IoT services. The testbed supports the simultaneous operation of several IoT services, with each service having its own physical sensor network and virtual cloud infrastructure at the application layer. It supports the IEEE 802.11 and LTE protocols for communication. The City of Things platform [31] is a testbed platform dedicated to Smart City experiments deployed in the city of Antwerp, Belgium. It allows the enforcement of city trials involving evaluating both the technological and user aspects. It comprises a complex wireless network infrastructure, enabling communications using heterogeneous IoT technologies: IEEE 802.1ac on 2.4 and 5 Ghz, DASH7 on 433 and 868 Mhz, BLE, IEEE 802.15.4, IEEE 802.15.4g and LoRa. The WISEBED [104] is a testbed for conducting WSN experiments. It includes a large, heterogeneous testbed, consisting of at least 9 geographically disparate networks that include both sensor and actuator nodes, and scaling in the order of thousands. It supports IEEE 802.11b/g and IEEE 802.15.4. Smaller testbed initiatives have also emerged, such as the OpenTestBed [105] which can be seen as a minimalistic testbed that can be easily and cheaply deployed for any IoT solution. It is a completely open-source hardware project, which, due to its simplicity, has been adopted by various institutions. A deployment of it with 80 motes (nodes) has been deployed in Paris, France with a total cost of less than 10.000 euros.

[103]: Teranishi et al. (2016), 'JOSE: An Open Testbed for Field Trials of Large-scale IoT Services'

[31]: Latre et al. (2016), 'City of Things: An Integrated and Multi-technology Testbed for IoT Smart City Experiments'

[104]: Chatzigiannakis et al. (2010), 'WISEBED: An Open Large-scale Wireless Sensor Network Testbed'

[105]: Munoz et al. (2019), 'OpenTestBed: Poor Man's IoT Testbed'

Summary: IoT testbeds are useful tools when it comes to the evaluation of IoT network technologies. They provide real-world environments for testing and validation, allowing researchers to evaluate the performance and behavior of IoT systems in authentic conditions. Testbeds offer opportunities for real-time monitoring and data collection, providing valuable insights into system behavior. However, they can be expensive to set up and maintain, limiting their accessibility. It is worth noting that the use of an IoT testbed implies constraints on the considered topology and scale. This may pertain to the distance between end-devices and their gateway, the maximum number of available IoT devices, and the environment which is typically indoors and not under the control of the researchers performing experiments. Testbeds may also have limited scalability compared to simulators, making it challenging to simulate large-scale deployments.

Simulators

Simulators are tools that aim at reproducing a physical system in a virtual environment. They are widely used in several domains, due to their ability to replicate physical processes without carrying the burden of deploying real material. In computer networks, discrete event simulators are extremely popular. A description of this paradigm is provided in the follows:

Discrete Event Paradigm: A Discrete Event system is a system where state changes (events) happen at discrete instances in time, and events are instantaneous (take zero time to happen). We assume that no other event happens between two consecutive events, *i.e.*, no changes occur in the system state [106]. In a network simulator, typical events can be the start or the end of a packet transmission, the expiry of a retransmission timer, etc.

[106]: Varga (2001), 'Discrete Event Simulation System'

We focus in what follows on two important aspects of discrete event simulators: The modeling of packet transmission and the energy consumption calculation.

(1) Packet Transmission Modeling: Let us suppose that node A sends a packet to node B through wireless network technology. When packet transmission occurs, the simulator calculates the time it will take to transmit the packet from A to B based on the available bandwidth, packet size, and other network parameters. The transmission is then scheduled by adding a new event to the event queue that represents the packet's arrival at the destination node. Then, to determine whether node B can receive the information without any bit errors, the simulator needs to compute the signal strength of the wireless transmission at node B. This is done through *Propagation Loss* models (also called path loss model).

Propagation loss models calculate the Rx (received) signal power according to the Tx (transmitter) signal power and the mutual Rx and Tx antennas positions, *i.e.*, the nodes positions. We present in what follows some of the most famous used propagation loss models:

- **LogDistance**: The log distance path loss model [107] assumes an exponential path loss over the distance from sender to receiver. It is designed for suburban scenarios.

[107]: Erceg et al. (1999), 'An Empirically based Path Loss Model for Wireless Channels in Suburban Environments'

[108]: Friis (1946), 'A Note on a Simple Transmission Formula'

- **Friis**: The propagation model proposed by Harald T. Friis [108] calculates quadratic path loss as it occurs in free space.
- **COST-Hata**: A model based on various experiments used to predict path loss in urban areas [109].
- **OkumuraHata**: The model presents path loss factors in Urban, Sub-Urban and open areas. It omits obstructions loss in the city environment [110].
- **TwoRayGround**: This model was initially developed by Rappaport [111]. It assumes a radio propagation via two paths: One ray is received directly while the other one reflects on the ground.

[109]: Damosso (1999), *Digital Mobile Radio towards Future Generation Systems: COST action 231*

[110]: Joseph et al. (2013), 'Urban Area Path Loss Propagation Prediction and Optimisation using HATA Model at 800MHz'

[111]: Rappaport et al. (2002), 'Wireless Communications: Past Events and a Future Perspective'

Note that in some simulators, different propagation loss models can be used in chains. The final Rx power takes into account all the chained models. It is also worth noting that propagation loss models are also often used in analytical models, to replicate the radio conditions of wireless channels.

(2) Energy Consumption Calculation: In most of the simulators, energy consumption is calculated using state machines, where the states are the different physical modes that a node can be on (sending Tx, receiving Rx, Idle, etc.). Each state is associated with a current consumption (in Amperes). Then, energy is calculated as follows:

$$E = \sum_{i \in S} (\alpha_i \times t_i) \times V \tag{3.1}$$

where:

- E: Energy consumption in Joules,
- S: Set of different physical states,
- α_i: Current consumption of state i in Amperes,
- t_i: Total time passed a state i in Seconds,
- V: Voltage in Volts.

It is worth noting that according to this modeling, simulators calculate only the energy induced by the transmission module. Other energy-intensive aspects such as sensing and processing [112] are not considered.

[112]: Martinez et al. (2015), 'The Power of Models: Modeling Power Consumption for IoT Devices'

We present in the following some of the most known discrete event network simulators in the community.

NS-3: The network simulator 3 [113], commonly called ns-3, is a C++-based discrete-event simulator that has been developed to provide an open and extensible network simulation platform for networking research and education. Due to its highly available documentation and the important set of network technologies it supports, it has become, along with ns-3, one of the most used simulators in the network community

[113]: Henderson et al. (2008), 'Network Simulations with the ns-3 Simulator'

OMNet++ OMNet++ [106] is a discrete event network simulator developed using C++ language by OpenSim company. It allows academic, educational and research-oriented commercial institutions to simulate computer networks and distributed systems. Due to the high range of Graphical User Interface (GUI) libraries for tracing and animating network scenarios, it has become one of the most used simulators in the networking community.

[106]: Varga (2001), 'Discrete Event Simulation System'

CupCarbon: CupCarbon [114] is a Smart City and IoT WSN simulator. It

[114]: Mehdi et al. (2014), 'Cupcarbon: A Multi-Agent and Discrete Event Wireless Sensor Network Design and Simulation Tool'

Table 3.2: Comparative Table of IoT Simulators.

Simulator	Scope	Last update	Type	Language	Evaluated Scale	Build-in IoT standards	Mobility	Target domain
NS-3 [113]	Network	2023	Discrete-event	C++	Large scale	802.15.4 LoRaWAN	Yes	Generic
OMNet++ [106]	Network	2022	Discrete-event	C++	Large scale	Manual extension	Yes	Generic
CupCarbon [114]	Network	2022	Agent based and discrete-event	Java /Custom Scripting	Large scale	802.15.4 LoRaWAN	Yes	Smart city
Cooja [116]	Network	2023	Discrete-event	C/Java	Large scale	All protocols supported by Contiki	Yes	Generic with focus on low-power sensors

can be considered both as a discrete event or an agent-based simulator[1]. It consists in a graphical tool that focuses on visualizing and designing algorithms that are required for monitoring and collecting environmental data. It allows two simulation environments: (i) one for the generation of natural events like fires and gas and (ii) one for the design of discrete event scenarios of WSNs. However, even though it is assumed possible to simulate 802.15.4 and LoRaWAN in the simulator, there is no reliable implementation of these communication protocols.

1: Agent-based models are models designed on a microscopic scale that simulate the simultaneous operations and interactions of several agents to recreate and predict the appearance of complex phenomena [115].

Cooja: Cooja [116] is a discrete-event simulator written in Java language, and which runs over the Contiki [54] OS. It is known for its high flexibility since it allows it to extend or replace some of its functions with new functionalities such as Operating Systems, sensor node platforms, radio transmission models, etc.

[116]: Osterlind et al. (2006), 'Cross-level Sensor Network Simulation with Cooja'

[54]: Dunkels et al. (2004), 'Contiki-a Lightweight and Flexible Operating System for Tiny Networked Sensors'

Table 3.2 is a comparative table between the presented network simulators, inspired from [10]. Note that the column denoted 'Evaluated Scale' refers to the scale at which evaluations using that simulator have been performed in the literature.

[10]: Kassab et al. (2020), 'A–Z survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations'

Some network simulators which are less used for IoT are available in the market. QualNet [117] is a discrete-event simulator that can support accurate simulations of large-scale networks involving diverse network components. It comprises application, routing, MAC, and physical layers in its network architecture. It supports 802.15.4 networks such as Zigbee. The well-known MATLAB [118] simulator is an efficient tool meant for designing and simulating dynamic and embedded systems *. It has the capability of modeling, simulating, and analyzing processes. It provides tools for modeling different network layers (physical, data link protocols, etc.). It can also help to solve optimization problems.

[117]: Dinesh et al. (2014), 'Qualnet Simulator'

[118]: Matlab (2012), 'Matlab'

Summary: Network simulators offer several advantages in IoT research and development, including the ability to simulate large-scale deployments and complex network conditions and cost-effectiveness. They also provide flexibility in parameter modification and detailed performance analysis. However, they have limitations, such as the challenge of achieving full realism and accuracy compared to real-world networks.

* https://www.mathworks.com/products/matlab.html

Simulators also require careful consideration of model accuracy and assumptions. Despite their drawbacks, network simulators remain valuable tools in IoT network evaluation.

Analytical Models

Analytical models are mathematical representations of an object or a system to predict its behavior. In the context of performance evaluation in IoT networks, analytical models are often used to model Key Performance Indicators (KPI)s according to the configuration parameters of IoT technologies and the workload. We present in what follows the KPIs we will focus on through this thesis, as well as some analytical models for each one:

1. **Throughput**: Attained throughput represents the overall speed of the network at conveying data or the data rate delivered to each IoT device. Ensuring high throughput is critical in applications involving the transmission of large amounts of data, such as video-surveillance. Different analytical models have been proposed in the literature for the throughput. We present some of them in what follows:

   [119] gives an analytical expression of packet throughput for 802.11b (Wi-Fi) networks with N end-devices of different bit rates competing for the radio channel. The resulting formula for the throughput X_f experienced by each end-device, in a network, the end-devices generate a connectionless UDP stream to a host behind the access point, is the following:

   $$X_f = \frac{T_s}{(N-1) \times T_f + T_s + P_c(N) \times t_{jam} \times N} \tag{3.2}$$

   where:

   - T_f: Transmission time of a "fast" host transmitting at a rate R,
   - T_s: Transmission time of a "slow" host transmitting at a rate r,
   - $P_c(N)$: Proportion of collisions experiences for each packet successfully acknowledged,
   - t_{jam}: Average time spent in collisions,
   - N: Number of competing end-devices,

   In some cases, instead of modeling the performance of a whole network, some efforts have been done to model the underlying mechanisms of IoT network technologies. For instance, Random Access (RA) procedure is a technique implemented in the MAC layer in NB-IoT to resolve the channel contention conflict among multiple user equipment (or end-devices)[2]. Thus, the procedure can affect the behavior of the whole network. [120] calculates the system throughput for the random access procedure in an NB-IoT network, according to the number of end-devices, the packet generation rate, the transmission number and the length of end-devices buffers. Considering all these aspects, they end up with the following equation for the throughput τ:

   $$\tau = \frac{\tau'}{T} \tag{3.3}$$

[119]: Heusse et al. (2003), 'Performance Anomaly of 802.11 b'

2: The RA procedure is initiated by user equipment in the idle state to request uplink radio resources required to send data.

[120]: Sun et al. (2017), 'Throughput Modeling and Analysis of Random Access in Narrowband Internet of Things'

with:

$$\tau' = \binom{N}{1}(1-P_e)P^{N-1} + \sum_{n=2}^{N}\left[\binom{N}{n}(1-P_e)^n P_e^{N-n}\right] \cdot \left[\sum_{\omega=1}^{W}\binom{n}{1}\frac{1}{W}\left(\frac{W-\omega}{W}\right)^{n-1}\right] \tag{3.4}$$

where:

- T: Cycle time,
- N: Number of end-devices,
- W: Number of delay parameters in backoff mechanism,
- P_e: Probability that a queue is empty,

[121]: LoRa Alliance Technical Commitee (2017), 'LoRaWAN 1.1 Specification'

According to its constructor (Semtech) in [121], the theoretical throughput T of LoRa networks can be modeled as the following formula:

$$T = SF\frac{BW}{2^{SF}CR} \tag{3.5}$$

where:

- SF: Spreading Factor,
- BW: Bandwidth,
- CR: Coding Rate

[122]: Yousuf et al. (2018), 'Throughput, Coverage and Scalability of LoRa LPWAN for Internet of Things'

[122] analyzes also the measured throughput in practice for LoRa, and notice a difference between the throughput at the radio and application level, due to protocol and application overheads that augment packet transmission time.

[123]: Mehrnoush et al. (2018), 'Analytical Modeling of Wi-Fi and LTE-LAA Coexistence: Throughput and Impact of Energy Detection Threshold'

3: It is an extension of the LTE wireless standard that operated on the unlicensed 5 GHz frequency band.

In some cases, it is necessary to model throughput (and more generally KPIs) in networks where different network technologies coexist. For instance, [123] studies the coexistence of Wi-Fi and LTE-Licensed Assisted Access (LTE-LAA)[3] in unlicensed bands (particularly 5 GHz). They propose an analytical model to estimate the throughput in such networks. The formula for the throughput T_w is the following:

$$T_w = \frac{P_{trw}P_{sw}(1-P_{trl})P_{size}}{T_E}r_w \tag{3.6}$$

where:

- P_{trw}: Transmission probability,
- P_{sw}: Success transmission duration,
- P_{trl}: LTE transmission probability,
- P_{size}: Data portion duration,
- T_E: Total average time,
- r_w: Wi-fi data rate.

We have seen that several parameters can affect the throughput in a network. Some of them are generic, such as the density (number of nodes), the data rate, etc. while others are more specific to network technologies. Indeed, for instance for LoRa, physical parameters such as the SF or the bandwidth can strongly affect the throughput. However, accurately calculating the throughput in a network requires considering additional factors, such as the radio environment (including factors like signal propagation, interference, and fading) or the nodes' positions.

2. **End-to-end Reliability**: The end-to-end reliability (a.k.a. success rate) is the ratio of the packets successfully received from all the sent packets. In applications where data is critical, such as alarms in a monitoring system, it is absolutely primordial to ensure a high packet delivery. Note that the packet delivery can be derived from the throughput and the initial application data rate.

3. **Packet Latency**: Latency is the time that a packet takes to transit from its source to its destination. High latency can be disastrous in real-time critical applications, such as controlling a drone for an emergency situation (natural disaster, etc.). Some models for the packet latency are presented in what follows:

[124] models the uplink latency in LoRaWAN networks, taking into account sub-band selection and the case of sub-band combining. The formula for the transmission time is, assuming that propagation and processing are negligible:

[124]: Sørensen et al. (2017), 'Analysis of Latency and MAC-layer Performance for Class A LoRaWAN'

$$T_{total} = T_{tx} + T_w \tag{3.7}$$

T_{tx} is the time needed to transmit LoRa symbols (or data), and which is in the function of SF, Coding Rate, etc. (see [125] for more details). T_w represents the waiting time, due to regulatory duty-cycling. It is calculated using a $M/M/c$ queue. This gives the following formula for the waiting time in sub-band i:

[125]: Semtech Corporation (2013), 'SX1272/3/6/7/8 LoRa Modem Design Guide, AN1200.13'

$$T_{w_i} = \frac{p_{busy,all}}{2\left(\sum_{i=1}^{c} \mu_i + \lambda\right)} \tag{3.8}$$

where:

- $p_{busy,all}$: Erlang-C probability that all servers are busy,
- μ_i: Service rate of queue i,
- λ: Arrival rate.

For NB-IoT, [126] model the data transmission delay per end-device for both downlink and uplink transmissions. The formula is the following:

[126]: El Soussi et al. (2018), 'Evaluating the Performance of eMTC and NB-IoT for Smart City Applications'

$$\begin{aligned}
&Delay_{ED}^{i} = TL_i \times \left[\frac{DataLen}{TBS(MCS, RBU)}\right], i \in \{DL, UL\}, with: \\
&TL_{DL} = RLDC \times t_{PDCCH} + t_D + RLSD \times t_{PDSCH} + t_{DUS} + RLUC \times t_{ULACK}, \\
&TL_{DL} = RLDC \times t_{PDCCH} + t_{DUS} + RLUS \times t_{PUSCH} + t_{UDS} + RLDC \times t_{DLACK},
\end{aligned} \tag{3.9}$$

where:

- $RLDC$: Number of repetitions of the physical downlink control channel (PDCCH),
- $RLDS$: Number of repetitions of the data on the physical downlink channel (PDCCH),
- $RLUS$: Number of repetitions of the data on the physical uplink channel (PDSCH),
- t_{PDCCH}: Transmission time needed needed to transmit the control information on the PDCCH,

- t_{PDSCH}: Transmission time needed needed to transmit one transport block on PDSCH,
- t_{PUSCH}: Transmission time needed needed to transmit one transport block on PUSCH,
- t_D: Cross subframe delay,
- t_{DUS}: Radio frequency tuning delay for switching from DL to UL,
- t_{DUS}: Radio frequency tuning delay for switching from UL to DL,
- $DataLen$: Data size per user,
- TBS: Transport block size,
- MCS: Modulation and coding scheme,
- RBU: Allocated resource block per user

As we have seen in the presented models, similarly to throughput, several parameters can affect packet latency. These parameters can be categorized into generic factors and technology-specific factors. Generic parameters that impact packet latency include contention and collisions. When multiple end-devices contend for the same channel resources, there is a higher likelihood of collisions, which can introduce latency. The level of contention and collisions in a network is influenced by factors such as the number of competing devices, the traffic load, and the access method employed. In addition to these generic parameters, the specific network technology being utilized can introduce additional factors that affect packet latency. Each network technology has its own set of low-level parameters that impact latency. For instance for LoRa, parameters such as the SF or channel bandwidth can influence the latency experienced by packets. It is important to consider both the generic and technology-specific parameters when analyzing packet latency in a network.

4. **Energy Consumption**: Energy is highly important in the IoT industry where end-devices often have limited power supply and are equipped with a battery. A related considered metric is power consumption, which represents the rate at which energy is consumed over a period of time. As mentioned before, energy (and power) consumption must be an obsession for every application featuring battery-powered end-devices. Due to its tremendous importance, modeling energy consumption in IoT networks has been the subject of substantial efforts from the community:

[112]: Martinez et al. (2015), 'The Power of Models: Modeling Power Consumption for IoT Devices'

[112] proposes a generic model for energy consumption in IoT networks. It presents a comprehensive model for the power consumption of wireless sensor nodes that takes into consideration the classical tasks of an IoT network: Communications, acquisition and processing. The model is only based on parameters that can be empirically quantified, once the platform (i.e., technology) and the application (i.e., operation conditions) are defined. The equation of the total current drained I_{DEV} in Ampere is the following:

$$I_{DEV} = \underbrace{\frac{\alpha N_S}{T_{RCD}}}_{Sampling} + \underbrace{\frac{\beta \tau(N_P)}{T_{RCD}}}_{Communication} + \underbrace{\frac{\gamma H(N_A)}{T_{MSG}}}_{Processing} + \delta \tag{3.10}$$

where:

- α: Charge per acquired sample,
- N_S: Number of acquired samples,
- T_{RCD}: Sampling period,
- β: Charge per instruction processing,
- τ: Complexity of processing function,
- N_P: Number of processed samples,
- γ: Charge per message transmission,
- N_A: Any radio-related parameter, for instance the number of retransmissions,
- H: Function of N_A, for instance $H(N_A) = N_A$ in case samples are sent in a unique message N_R times,
- T_{MSG}: Transmission period,
- δ: System activity current,

The current drained can then be used to determine the power consumption of the network, using the following equation: $P_{DEV} = I_{DEV} \times V_{BUF}$, where P_{DEV} is the power consumption of the device and V_{BUF} the voltage in its terminals. Then, the power consumption can finally be used to determine the battery lifetime of the end-devices, following the equation $L = \frac{V_{BAT} \times C_{BAT}}{P_{DEV}}$ where L is the battery lifetime in hours, V_{BAT} is the battery voltage in Volts, C_{BAT} is the battery capacity in Amp.hours (assuming a 100% efficiency between the battery and the appliance).

[127] presents a model to estimate the dissipated energy for WSNs in general. They affirm that of all the sensor node components, the most energy-consuming part is the RF module. They use a state machine to calculate the energy associated with the communication, where each physical state of the RF module (transmitting, receiving, idle and sleeping) is associated with current consumption. Not surprisingly, the conclusion of the paper is that the RF module and the controller should be in idle state as long as possible when they are not active.

[127]: Srbinovska et al. (2017), 'Energy Consumption Estimation of Wireless Sensor Networks in Greenhouse Crop Production'

The same approach is followed in [128], but specifically for LoRaWAN end-devices. Moreover, they distinguish between acknowledged and unacknowledged transmission. They consider the following states related to transmission activities:

[128]: Casals et al. (2017), 'Modeling the Energy Performance of LoRaWAN'

- T_{wu}: Wake up,
- T_{pre}: Radio preparation,
- T_{tx}: Transmission,
- T_{w1w}: Wait 1st window,
- T_{rx1w}: 1st receive window,
- T_{w2w}: Wait 2nd window,
- T_{rx2w}: 2nd receive window,
- T_{off}: Radio off,
- T_{post}: Turn off sequence,
- T_{seq}: Sleep.

[129] exhibits a Markov chain-based model for the energy consumption of NB-IoT networks. The model is used to estimate the energy consumption of an NB-IoT device sending periodic uplink reports using the control plane procedure. The model has been validated

[129]: Andres-Maldonado et al. (2019), 'Analytical Modeling and Experimental Validation of NB-IoT Device Energy Consumption'

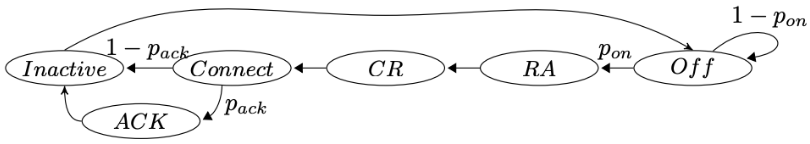


Figure 3.1: Markovian Chain for NB-IoT Energy Consumption Modeling [129].

using a base station emulator.

[130]: Serrano et al. (2014), 'Per-frame Energy Consumption in 802.11 Devices and its Implication on Modeling and Design'

[130] gives a model of the power consumption of a given list of 802.11 (Wi-Fi) devices by experimentation, with a highlight on the influence of parameters like MCS, frame size and generation rate on the energy consumption. They introduce the notion of *cross-factor*, which refers to the cost of the data frame crossing the protocol stack (OS, driver and Network Interface Card (NIC)) and which is independent of the frame size. The power consumed by an 802.11 device is given by the following equation:

$$P = \rho_{id} + \rho_{tx}\tau_{tx} + \rho_{rx}\tau_{rx} + \rho_{xg}\lambda g + \rho_{xr}\lambda_r \tag{3.11}$$

where:

- ρ_{id}: System activity in Idle state,
- ρ_{tx}: Consumed energy while sending,
- τ_{tx}: Time on Air percentage while sending,
- ρ_{rx}: Consumed energy while receiving,
- τ_{rx}: Time on Air percentage while receiving,
- ρ_{xg}: Cross-factor for sending,
- λ_g: Sending frame rate,
- ρ_{xr}: Cross-factor for receiving,
- λ_r: Receiving frame rate,

We have seen different models for energy consumption in networks. From more generic ones that can be applied to any network technology and that try to capture the energy consumed by the different tasks of IoT (communication, sensing, processing, etc.), to more specific ones that can be applied to precise network technologies, the purpose is the same: Evaluating the energy consumption induced in a whole network. Note that, when coming to the selection of an IoT network technology, we tend to neglect the energy consumption induced by sensing and processing, since they can be considered regardless of the network technology. Thus, in this thesis, we focus only on the energy consumption induced by transmissions.

Summary: As we have seen in this subsection, using mathematical models for calculating KPIs can be very challenging. It is indeed complex to mathematically capture and consider all the parameters that can affect a given KPI. In addition, each KPI would typically need a specific model, with often restrictive hypotheses about the considered network. These reasons may lead researchers to use real experiments with testbeds but at a higher cost, or simulators in a cheaper way. Indeed, many simulators aim at reproducing the different layers of a network, they can thus be able to capture parameters easier than mathematical models, and with less restrictive hypotheses than analytical models. Indeed, most of the presented models assumed specific transport protocols (*e.g.,* UDP), traffic behaviors (different rates for the end-devices) or even coexistence

Table 3.3: Summary of Testbeds, Simulators and Analytical Models.

Evaluation Tools		KPIs Packet Throughput	Packet Latency	Energy Consumption
Testbeds	FIT IoT-Lab	✓	✓	✓
	Smart Santander	✓	✓	✓
	w-iLab.t	✓	✓	✓
Simulators	NS-3	✓	✓	✓
	OMNeT++	✓	✓	✓
	CupCarbon	✓	✓	✓
	Cooja	✓	✓	✓
Analytical Models	(Heusse et al., 2003) [119]	✓	-	-
	(Sun et al., 2017) [120]	✓	-	-
	(LoRa Alliance, 2017) [121]	✓	-	-
	(Yousouf et al., 2018) [122]	✓	-	-
	(Mehrnoush et al., 2018) [123]	✓	-	-
	(Sorensen et al., 2017) [124]	-	✓	-
	(Semtech Corp., 2013)[125]	-	✓	-
	(El Soussi et al., 2018) [126]	-	✓	-
	(Martinez et al., 2015) [112]	-	-	✓
	(Srbinovska et al., 2017) [127]	-	-	✓
	(Casals et al., 2017) [128]	-	-	✓
	(Andres-Maldonado et al., 2019) [129]	-	-	✓
	(Serrano et al., 2014) [130]	-	-	✓

between several network technologies. A generalization to other transport protocols, traffic workloads or network technologies would typically require to design new models.

Table 3.3 provides a summary of the testbeds, simulators, and analytical models discussed, highlighting the specific KPIs they target. Notably, both testbeds and simulators offer the advantage of not being limited to a single KPI, unlike analytical models. While this may sacrifice some precision, it enables researchers to conduct more comprehensive analyses more easily compared to modeling each individual KPI separately.

Conclusion

By employing testbeds, simulators and analytical models, stakeholders can now make informed and objective choices based on specific criteria and objectives. These approaches enable the evaluation and comparison of network technologies, considering factors such as performance, cost, energy consumption, etc. As a result, several scientific works for the evaluation, selection, comparison and configuration of IoT networks have emerged in the literature. We present some of them in the following.

3.2.2 IoT Network Technologies Comparison Studies

A number of works have conducted performance studies in a bid to compare the efficiency of two or more network technologies in supporting an IoT application. Typically, they consider a specific scenario and evaluate the associated performance using simulations or real experiments. In [131] the authors assess the relative merits of NB-IoT, SigFox, and LoRaWAN in covering the needs of smart water grids using the simulator ns-3. While they conclude own the superiority of NB-IoT, their study does not take into account major KPIs such as latency, cost, range, and energy consumption. [132] compares Wi-Fi HaLow, LoRaWAN and NB-IoT for smart city applications, using simulation. In [133], the authors use simulations to compare the coverage and capacity of SigFox, LoRaWAN and NB-IoT at meeting the needs of a large-scale IoT deployment. While all technologies were found able to cover most of the needs in terms of coverage for outdoor communications, their results show that NB-IoT, and to a lesser extent Sigfox, outperform the others for indoor communication. However, their results do not consider energy- and delay-related KPIs. All these works do not provide any generic tool for the selection of network technologies. To fulfill this gap, we think that additional steps are required, such as the modeling of the targeted IoT application and the network alternatives, the application of an evaluation framework to evaluate the KPI values obtained with the network alternatives, and the use of a comparison method to rank the different network alternatives and to identify the best one. In [12], the authors propose a 2-step methodology to guide IoT users to choose the appropriate network technology for their needs. First, they use a questionnaire to eliminate network technologies based on the mismatch between the application requirements and the network technology characteristics. Then, they propose an evaluation of the main cost components to find the most economical network technology. Overall, their solution can be viewed as a solid step towards an automatic selection method for the choice of an IoT network technology. However, the considered values for KPIs are constant (when they should vary with the scenario under consideration) and the relative merits of IoT network technologies are compared only through their financial cost.

[12]: Vannieuwenborg et al. (2018), 'Choosing IoT-connectivity? A guiding Methodology based on Functional Characteristics and Economic Considerations'
[134]: Bari et al. (2007), 'Multi-Attribute Network Selection by Iterative TOPSIS for Heterogeneous Wireless Access'
[135]: Bari et al. (2007), 'Automated Network Selection in a Heterogeneous Wireless Network Environment'
[136]: Senouci et al. (2016), 'TOPSIS-based Dynamic Approach for Mobile Network Interface Selection'
[137]: Tzeng et al. (2011), *Multiple Attribute Decision Making: Methods and Applications*
[138]: Gazis, Vangelis, et al. (2005), 'Toward a Generic" Always Best Connected" Capability in Integrated WLAN/UMTS Cellular Mobile Networks (and Beyond)'
[139]: Wang et al. (2009), 'Best Permutation: A Novel Network Selection Scheme in Heterogeneous Wireless Networks'
[140]: Hou et al. (2006), 'Vertical Handover-Decision-Making Algorithm using Fuzzy Logic for the Integrated Radio-and-OW System'
[141]: Zhang (2004), 'Handover Decision using Fuzzy MADM in Heterogeneous Networks'
[142]: Stevens-Navarro, Enrique, et al. (2008), 'An MDP-based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks'
[143]: Ning, Zhaolong, et al. (2014), 'Markov-based Vertical Handoff Decision Algorithms in Heterogeneous Wireless Networks'
[144]: Cesana et al. (2008), 'Game Theoretic Analysis of Wireless Access Network Selection: Models, Inefficiency Bounds, and Algorithms'
[145]: Liu, Bin, et al. (2014), 'AHP and Game Theory based Approach for Network Selection in Heterogeneous Wireless Networks'
[146]: Gai et al. (2012), 'Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations'
[147]: Henri, Sébastien, et al. (2018), 'Multi-Armed Bandit in Action: Optimizing Performance in Dynamic Hybrid Networks'

In terms of decision support, studies have mostly focused on the dynamic interface selection, a.k.a. Vertical Handoff (VHO) with the goal of favoring the performance of end-users [134–136]. These studies naturally lead to dynamic multi-criteria decision problems where a utility function must be cautiously devised using algorithms such as Simple Additive Weighting (SAW), Weighted Product Methods (WPM), and Technique for Order Preference Similarity to Ideal Solution (TOPSIS) [137]. To handle the exploration-exploitation dilemma inherent to the dynamic selection of interfaces, researchers have resorted to various mathematical approaches such as combinatorial optimization [138, 139], fuzzy logic [140, 141], Markov Decision Process (MDP) [142, 143], game theory [144, 145], and ML techniques such as the multi-armed bandit framework [146, 147]. All these works suppose that a heterogeneous wireless network has been put in place and configured so that probing tests can be performed before selecting the most adequate network interface for each end-device. This is in contrast with the fact that for the static selection of the network technology during the design phase, the network has not been yet deployed, nullifying the possibility of collecting performance probing.

We observe that, despite the abundance of IoT network research, there is a relative lack for developing a reproducible and robust approach to systematically analyze the matching between an application and a network solution and its scalability. Most papers are indeed either restricted to the study of a single communication technology or alternately, to only one application. Moreover, unlike fields like linear algebra and image processing, the IoT community lacks a reproducible approach to assess the performance of an IoT network technology in a well-defined usage context.

3.2.3 IoT Network Technologies Configuration Studies

The MAC parameters of the 802.15.4 standard are often the ones that we try to optimize. For example, the authors of [148] propose to use a neural network to find the optimal values of these MAC parameters to maximize throughput and minimize latency. [149] propose a solution based on a supervised learning algorithm to estimate the optimal of IEEE 802.15.4 MAC parameters using simulation to predict the optimal parameters that improve best the end-to-end transmission delay.

[148]: Al-Kaseem et al. (2017), 'A New Intelligent Approach for Optimizing 6LoWPAN MAC Layer Parameters'

[149]: Aboubakar et al. (2020), 'An Efficient and Adaptive Configuration of IEEE 802.15. 4 MAC for Communication Delay Optimisation'

Regarding Wi-Fi, an important number of parameters can be considered for optimization. In [150], the authors propose an online-learning-based solution based on network load and channel conditions to achieve high throughput in Wi-Fi-based topologies. The considered parameters are the channel bandwidth, the MCS values and the number of Multiple Input Multiple Output (MIMO) antennas. [151] employ a deep learning approach to search for optimal configuration of contention window for improving the throughput, the transmission delay and packet retransmission rate. They use simulation to generate data for several applications with different numbers of nodes, short interframe space, data transmission rates, etc. In [152], they provide an algorithm to adapt the amount of 802.11n aggregation by Wi-Fi stations according to the level of congestion in the network, with the objective of optimizing the QoS and the energy efficiency.

[150]: Karmakar et al. (2020), 'An Online Learning Approach for Auto Link-Configuration in IEEE 802.11 AC Wireless Networks'

[151]: Chen et al. (2020), 'Contention Resolution in Wi-Fi 6-enabled Internet of Things based on Deep Learning'

As we can see, most of these methods rely on analytical models, which may induce restrictive assumptions on the studied network. Also, they are restricted to a very network technology and for specific KPIs. There is a need for a generic method that can be applied for any network technology, for any application, and according to any KPIs. To do so, relying on simulation may be an interesting solution since it provides abstraction and allows the calculation of several KPIs.

3.3 Conclusion

We have presented in this chapter a state of the art of evaluation of IoT network technologies. We began by presenting field approaches that IoT architects usually refer to before the deployment of their IoT solutions. Then, we presented the different scientific approaches for the evaluation of IoT network technologies, which are experimentation, analytical models and simulation. Then, we provided an overview of the related work of the selection and configuration of IoT network technologies.

As we have seen, the issues of evaluating, selecting and configuring IoT network technologies have been subject to important effort among the research community. The proposed approaches rely on different evaluation approaches, namely experimentation, analytical models and simulation. However, there is no generic framework or method that aims at supporting IoT architects during the deployment phase of their IoT solution. Indeed, most of the presented methods are either dedicated to a given application or to a given network technology. Furthermore, the proposed approaches require some expertise in networking or programming. Thus, they may be inaccessible for IoT architects who may lack this expertise.

We present in the next chapter the first contribution of this thesis, which is a generic framework for evaluating the performance of an IoT network technology for a given application, according to different KPIs.

IoT Network Technologies Modeling & Evaluation 4

4.1 Introduction

New network technologies have emerged and continue to evolve to accommodate the specific needs of IoT traffic. Solutions for long-range low-power communications include LoRaWAN, Sigfox, Wi-Fi HaLow, and NB-IoT, while short or medium-range communication technologies comprise Zigbee and Wi-Fi. The profusion and diversity of these network technologies which are regularly evolving can be confusing and disorienting to IoT users but also to IoT architects in charge of selecting and configuring them. In such a context, making decisions about the selection of network technology can be a complex and risky task. These decisions can affect the entire initiative, both for the achievement of the objectives set and for its future. Choosing the wrong IoT network is often not an option as budget, capacity and performance can be highly constrained. Ultimately, the decision generally comes down to the best trade-off between cost, range, throughput and battery lifetime for the targeted IoT application. These decisions need visibility and forecasts that are difficult, time-consuming and error-prone if done by hand.

Moreover, it is often a difficult task to model an IoT application and a network technology. This is due, on the one hand, to the diversity of parameters that can be considered in this regard. On the other hand, it is challenging to capture the IoT architect's needs when modeling an IoT application. For instance, when deploying an IoT solution, some architects may be more inclined to focus on the radio conditions of the environment, while others will aim attention to the number of end-devices and their positions.

In this chapter, we focus on the modeling and evaluation of IoT network technologies for a given application. We propose a framework consisting of modeling and associated tools to assist IoT architects in the evaluation of the ability of a network technology to meet the specific needs brought by a real-life IoT application. This framework abstracts the communication requirements of any IoT application and captures the specificity of any IoT network technology. It also defines and computes network- and energy-related Key Performance Indicators (KPI) that are used to assess the performance of a setting for a given topology and workload. To show the relevance of the method, we present the results of its application to three case studies inspired by IoT real-life. Finally, we discuss how IoT architects can exploit the outcome of this evaluation framework.

The remainder of the chapter is organized as follows. In Section 4.2, we formalize the problem of evaluating the performance of a network technology and its configuration for a given application. Section 4.3 describes the proposed application-driven evaluation framework. We explore three real-life inspired case studies to showcase the potential of our framework in Section 4.4. A discussion on the obtained results and guidelines on how they can be used are provided in Section 4.5. Finally, Section 4.6 concludes this chapter and gives some potential perspectives.

4.2 Problem Formulation

[153]: Rozanski et al. (2011), *Software Systems Architecture: Working with Stakeholders using Viewpoints and Perspectives*

In general, systems stakeholders have non-functional requirements such as performance, security, or scalability [153] which have to be taken into account and implemented as properties when designing systems. Here, we focus on the performance and scalability of the physical IoT network, which denotes the ability of the system to support the required performance profile and to handle increased processing volumes in the future if required.

We formulate the evaluation problem of an IoT network technology for a given application as follows. Let us consider:

- An IoT application A, with its set R of characteristics and communication requirements,
- a network technology T,
- a network configuration C for T,
- an evaluation tool E,
- a set K of key performance indicators or KPI that characterize the behavior of an application A on the network technology T configured according to C,

The evaluation problem consists in determining the values of K, resulting from the evaluation of the the network technology T and its associated configuration C on the application A, using the evaluation tool E. To address this problem, we present our modeling and evaluation framework in the next section.

4.3 Modeling & Evaluation Framework

In this section, we describe our application-based IoT network performance evaluation framework. It consists of two types of inputs, a tool and a set of outputs. These four building blocks are: (i) An application scenario specification (input), (ii) network setup characteristics (input), (iii) an evaluation tool and (iv) a set of key performance indicators (output) to assess the performance of the IoT network technology on the selected scenario. Figure 4.1 gives an overview of our evaluation framework, highlighting its inputs and outputs. We detail each of these components in the sections below.

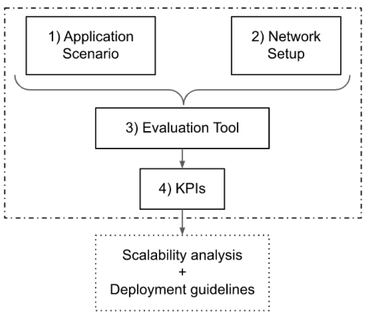


Figure 4.1: Overview of the Evaluation Framework Building Blocks.

4.3.1 Application Modeling

The principle of the application modeling and abstraction consists in characterizing the load imposed on the network by the application scenario over time. This load is a function of the number of end-devices and the individual traffic they are going to exchange. To study the evolutivity and scalability of the solution, the minimal and maximal values expected for the different selected parameters have to be specified.

Then the factors that characterize the environmental conditions which also impact wireless communications performance are added. The specific communication requirements of an IoT application are defined by the end-devices that are communicating, by the workload they impose on the network and their physical environment as defined below. Concerning the end-devices, we focus on:

(i) The number of end-devices that will be connected,
(ii) The maximal numbers of end-devices that could be ultimately connected,
(iii) The battery capacity of the end-devices.

For the workload we have:

(iv) The traffic direction (downstream and/or upstream traffic),
(v) The message size,
(vi) The message frequency,
(vii) The maximal message frequency, that could be ultimately submitted.

We propose to classify the IoT traffic types by: (i) Their direction, which can be upstream (from end-devices to gateways or the cloud) or downstream (from the cloud or gateways to end-devices) and (ii) their profile: Periodic or stochastic (for sporadic or bursty traffic). The periodic traffic corresponds to a fixed data rate, while the stochastic traffic has a variable rate. Although some applications have bidirectional traffic, we observe that a majority of IoT applications have unidirectional data flows. Table 4.1 categorizes the different IoT traffic types.

Table 4.1: Traffic Types Characteristics.

Traffic type	Traffic profile	Traffic direction	Examples
1	Periodic	Upstream	Telemetry, Geolocation...
2	Periodic	Downstream	Webcast, Virtual Reality...
3	Stochastic	Upstream	Video-surveillance, Cloud gaming...
4	Stochastic	Downstream	Notifications, Alerts, Remote commands...

We model the type of physical environment and deployment characteristics as follows:

(viii) The deployment scope, which is represented by the maximum distance expected between two end-devices,
(ix) The environment, which defines the radio conditions in which the IoT application is deployed,
(x) The expected lifetime of the deployed IoT solution must also be defined.

For the environment, we consider two cases: Indoor or outdoor, where the latter can be either (a) rural, (b) suburban or (c) urban. Inspired by [154], we propose to associate a propagation (path loss) model to each

[154]: Stoffers et al. (2012), 'Comparing the ns-3 Propagation Models'

environment type to characterize this environment as shown in Table 4.2.

Table 4.2: Environments and their Propagation Model.

Environment Type	Propagation Model [154]
Indoor	HybridBuildings
Outdoor Rural	OkumuraHata
Outdoor Urban	COSTHata
Outdoor Suburban	LogDistance

We consider that the knowledge of the ten parameters (i) to (x) is sufficient to characterize the targeted application scenario in the case of static end-devices for the network decision process. In the case of mobile end-devices, the mobility model will have to be specified. We do not consider this type of scenario in this thesis. Figure 4.2 illustrates the set of parameters considered to model and capture the main characteristics of an IoT application deployed in a rural environment.

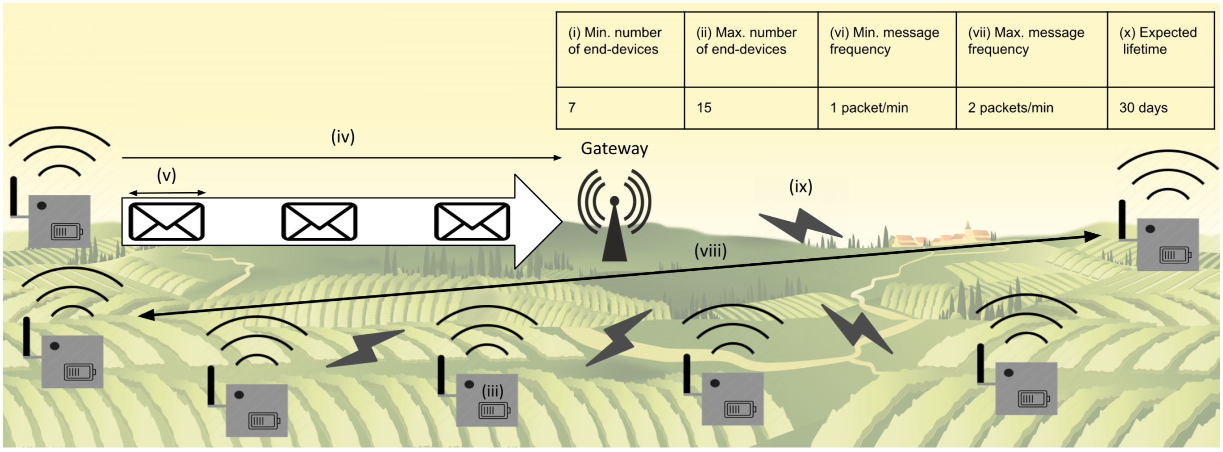


Figure 4.2: Abstraction of IoT Application whose characterizing parameters are: (i) Min. number of end-devices, (ii) Max. number of end-devices, (iii) End-device battery capacity, (iv) Traffic direction, (v) Message size, (vi) Min. message frequency, (vii) Max. message frequency, (viii) Deployment scope, (ix) Radio environment and (x) Expected lifetime.

4.3.2 Network Setup

For each network technology, we identify generic and specific parameters. Generic parameters, such as the maximum data rate (or bandwidth), the frequency band, and the topology type characterize any network technology. Specific parameters are dependent on each network technology. For instance, in the case of LoRaWAN, the specific parameters include the Spreading Factor (SF), the coding rate and the type of traffic (unconfirmed or confirmed). We note that some parameters are easily configurable by architects or by software (*e.g.*, SF for LoRaWAN) while others tend to be less tunable or simply out of reach for the architects (*e.g.*, the transmission power for LoRaWAN or Modulation and Coding Scheme (MCS) in Wi-Fi).

The generic parameters which are common to all the network technologies are: (i) The data rate, which is the theoretical maximal amount of data that can be sent per unit of time, (i) the frequency band, which is the frequency where the radio waves operate on and (iii) the topology type, which can be star or mesh.

We describe in what follows some of the specific parameters of LoRaWAN, Wi-Fi (802.11ac), 802.15.4 (6LoWPAN), Wi-Fi HaLoW (802.11ah) and 5G mmWave.

- **LoRaWAN**:
  - **Spreading Factor (SF):** Determines the speed at which the signal frequency changes across the bandwidth of a channel. The higher the spreading factor the lower the data rate.
  - **Coding Rate (CR):** An indication of how much of the data stream is actually being used to transmit usable data.
  - **Cyclic Redundancy Check (CRC)** An error-detecting code commonly used in networks to detect accidental changes in the transmitted data.
  - **Type of traffic:** Determines whether the data is sent with or without an acknowledgement. It can therefore be confirmed or unconfirmed, respectively.
- **Wi-Fi (802.11ac)**:
  - **Number of spatial streams:** Determines the number of streams where coded data signals can be sent and received independently.
  - **Packet aggregation:** Determines whether packet aggregation, which is the process of joining multiple packets together into a single transmission unit, is enabled or disabled.
  - **Guard Interval:** Is the space between symbols (characters) being transmitted. It can either be short (0.4 µS) or long (0.8 µS).
  - **Modulation and Coding Scheme (MCS):** An index based on several parameters of a Wi-Fi connection between two stations. Namely, for 802.11ac, it depends on the modulation type, the coding rate, the number of spatial streams, the channel width, and the guard interval.
- **802.15.4 (6LoWPAN)**:
  - **Number of frame retries:** It is the number of the retransmissions limit when there is no acknowledgement received before dropping the packet.
  - **Carrier Sense Multiple Access (CSMA) backoff:** The number of times that the node stays in the backoff stage after unsuccessful channel sensing.
  - **Maximal backoff exponent:** Maximal random interval before sensing the channel.
  - **Minimal backoff exponent:** Minimal random interval before sensing the channel.
- **Wi-Fi HaLow (802.11ah)**:
  - **Guard Interval**: Same for Wi-Fi.
  - **Beacon interval**: Interval of time at which the access point (gateway) broadcasts its beacon frames (used for controlling, etc.).
  - **Number of Random Access Window (RAW) groups**: Number of groups at which the stations are split on. Each group has one or more slots over which the stations belonging to that group are evenly split. During a RAW slot, only the stations that belong to that slot are allowed to access the channel.

► **5G mmWave**

- **Numerology**: It refers to the subcarrier spacing and the length of the time interval used for transmitting data. The higher the numerology, the higher the number of slots per subframe, *i.e.*, the more data can be sent during that subframe.
- **Hybrid Automatic Request (HARQ)**: When active, it allows the detection of error in transmission, and the receiver sends a retransmission request to the sender when needed.
- **Radio Ling Control Acknowledged Mode (RLC-AM)**: When active, the receiver acknowledges the correct reception of the packet by sending an Acknowledge (ACK) message to the sender for every successfully received data packet.

A summary of this modeling is presented in Table 4.3.

Table 4.3: Specific Parameters of some Network Technologies models.

Network technology	List of specific parameters
LoRaWAN	• SF $\in [7; 12]$ • CR $\in [1; 4]$ • CRC $\in \{0, 1\}$ • Type of traffic $\in$ {unconfirmed, confirmed}
5G mmWave	• Numerology $\in [1; 5]$ • HARQ $\in \{0, 1\}$ • RLC-AM $\in \{0, 1\}$
Wi-Fi HaLow	• MCS $\in [0; 9]$ • Spatial streams $\in [1; 3]$
Wi-Fi	• MCS $\in [0; 9]$ • Spatial streams $\in [1; 3]$ • Packet aggregation $\in \{0, 1\}$
802.15.4	• Min. Backoff exponent $\in [0; 7]$ • Min. Backoff exponent $\in [3; 8]$ • Max. CSMA backoff $\in [0; 5]$ • Max. frame retries $\in [0; 7]$

4.3.3 Evaluation Approach

As stated in Chapter 1, the performance evaluation could be realized through different approaches: Experimentation, simulation or analytical modeling. These approaches have their own advantages and drawbacks. Depending on their requirements, users can choose between one or several of these evaluation approaches. If real performance results are expected, experimentation will be preferred. For example, the architects will run a real test if they need to precisely estimate the battery power consumption under a given workload. If a scalability analysis is considered, analytical modeling or simulation is very likely more appropriate. IoT architects will often conduct studies combining several evaluation approaches to get precise at-scale results.

4.3.4 IoT-relevant KPIs

Now, we propose to focus here on the following KPIs as evaluation outputs: (i) Attained throughput, (ii) message latency, (iii) reliability, (iv) power consumption, (v) energy efficiency ratio, (vi) battery lifetime, (vii) scalability index and (viii) cost.

As a recall, attained throughput, message latency and reliability are classical networking performance parameters. Attained throughput represents the overall speed of the network at conveying data or the data rate delivered to each IoT device. message latency is the time that a packet takes to transit from its source to its destination. The reliability (a.k.a. success rate) is the ratio of the packets successfully received from all the sent packets. Note that from the IoT application perspective, a message is the fundamental data unit while the packet is the classical network data unit. There may be several packets in one application message. But for the sake of simplicity and without loss of generality we assume a message corresponds to one packet here.

Energy is highly important in the IoT industry where end-devices often have limited power supply and are equipped with a battery. Power consumption represents the rate at which energy is consumed over a period of time. It can be measured on the overall network or on each IoT end-device. In this work, we define the energy efficiency ratio as the number of bytes that each transmitter can successfully transmit to the receiver using a single joule of energy. The higher this quantity, the more energy efficient the IoT technology is. The battery lifetime gives an indication of the IoT system's lifetime without recharging batteries. Note that we focus here only on energy consumption due to the transmission costs. Sensing/actuating energy consumption is not considered.

We call the "scalability index" the maximum number of devices that can be connected to a single gateway without deteriorating the performance in terms of IoT metrics. Cost represents the financial cost for deploying and maintaining the considered network technology given a selected number of gateways and network configurations for the lifetime of the project.

Note that other KPIs such as security or environmental impact can also be included if the case study of interest requires it. For security, a common way is to consider the robustness of the employed encryption algorithm. Regarding the environmental impact, one way of evaluating that involves analyzing the environmental impacts at each stage of the end-devices' lifecycle, including raw material extraction, manufacturing, use phase, and end-of-life disposal. This would naturally induce much more effort as it would need to clearly specify the type of end-device as well.

4.4 Examples of Application

In this section, we apply our framework to two different use cases: Telemetry and video-surveillance. We evaluate the adequacy and the performance of the Wi-Fi and LoRaWAN network technologies, for these use cases. We use simulation in the Network Simulator 3 (ns-3) environment as the evaluation tool. We used the release 3.33 of ns-3 for

Wi-Fi, and a code patch not integrated into the official version of Network Simulator 3 but widely used by the research community to evaluate LoRaWAN [155].

[155]: Magrin et al. (2017), 'Performance Evaluation of LoRa Networks in a Smart City Scenario'

First, we consider the following telemetry use-case: An IoT architect has to design a WSN-based service to count passengers in urban trains, where sensors are placed over each door. The counting service will operate in real-time to optimize the passenger flows. A typical train will have up to twenty wagons and a length going up to 1000m. In this case, the key questions the IoT architect would have is how many gateways should be installed as well as how frequently messages can be exchanged with 99% reliability.

The second use case is related to video-surveillance. We consider a large event gathering a large crowd (Olympic games, trade fairs, concerts, etc.) where a camera-based surveillance system is needed. For mobility, installation time and logistic reasons, the only possible solution is to adopt a wireless connectivity. This means that cameras have to be placed at specific locations while being self-powered with batteries. Video frames will then be transmitted to a server through a wireless network. A crucial challenge for IoT architects is to know how long will the batteries last, depending on the number of cameras and the video quality. Moreover, it would be interesting for them to know how many cameras should be placed, and at what distance from each other to avoid collisions. The energy efficiency ratio is also studied as it represents the energy consumption behavior in an interesting way.

All these answers have a critical impact, especially from the financial viewpoint, since they may give the maximum number of gateways, sensors or cameras that can be installed, they can also inform on how often the batteries will have to be changed, etc. We explore these questions in the following, using our framework to analyze costs and scalability. In order to provide actionable results, we vary parameters that are critical from the application perspective (message size, message period, etc.), as well as the number of end-devices.

Note that in what follows, the modeling of the IoT applications does not include the expected lifetime (the whole deployment time) of the IoT solution, as we do not compare several network technologies. We also do not consider the cost KPI, since we only consider scenarios with one gateway.

4.4.1 Case study A: Telemetry on LoRaWAN

This example is devoted to the case of a telemetry application deployed over LoRaWAN. The sensors collect data before exchanging them to the gateway for further processing.

The application scenario is defined as the following: We let the number of sensors vary from 1 to 15,000 with a deployment scope of 3,000 meters from the gateway. Each sensor is equipped with a battery of 2,400 mAh capacity powered by 3.3 V (such a battery is used in [128]). The traffic type corresponds to type 1 (periodic and upstream) of Table 4.1. Regarding the workload, the size of packets (a.k.a. payload) is set to 23 bytes unless specified otherwise, and we consider three possible periods

[128]: Casals et al. (2017), 'Modeling the Energy Performance of LoRaWAN'

Application modeling	Parameters	Case A
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	1 15,000 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream 23 0.001 0.003
Environment	• Type • Scope (meters) • Expected lifetime (days)	Suburban 3000 N/A

Table 4.4: Application Modeling of Case A.

for the rate at which sensors generate their packets: 300, 600, and 900 seconds. We consider a sub-urban radio environment. These parameters can be found in Table 4.4. For the network setup, we consider that the sensors communicate using LoRaWAN, on the 868 MHz frequency band with a bandwidth of 125 KHz. To evaluate the influence of the SF over the KPIs, we consider two of its value: 7 and 9. The network has a star topology with one gateway. As mentioned before, we used simulation as the evaluation tool.

[155]: Magrin et al. (2017), 'Performance Evaluation of LoRa Networks in a Smart City Scenario'

To evaluate the performance parameters for this example, we run simulations of 3600 seconds using ns-3. Although the official release of ns-3 does not include methods to estimate the energy costs incurred by LoRaWAN communications, Magrin et al. provide an ns-3 module [155] to do so. The power consumption of the NIC is obtained thanks to a state machine whose states and associated drawn currents are given in Appendix (Table 4.5). Having set this module, we are then able to obtain the KPIs for this network technology.

State	Drawn current value (mA)
Tx	77
Rx	28
Idle	1
Sleep	0.015

Table 4.5: Drawn current values for each state of the machine state used in ns-3 simulations to evaluate the power consumption of LoRaWAN communications.

Figure 4.3 shows the results provided by our framework for case study

(a) Reliability (b) Energy Efficiency Ratio

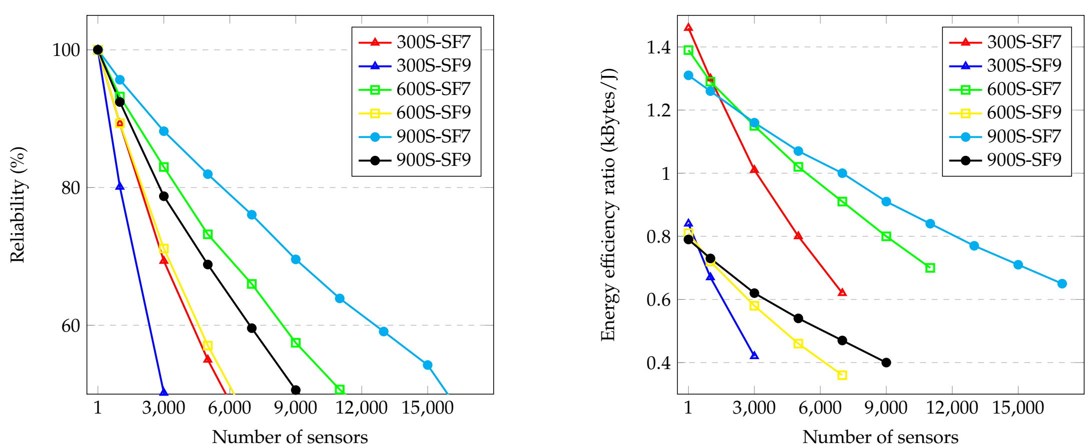


Figure 4.3: Case study A: KPIs for Telemetry on LoRaWAN.

A, while Figure 4.4 represents an illustration of our framework on the Case Study A, with an instantiation of the four building blocks.

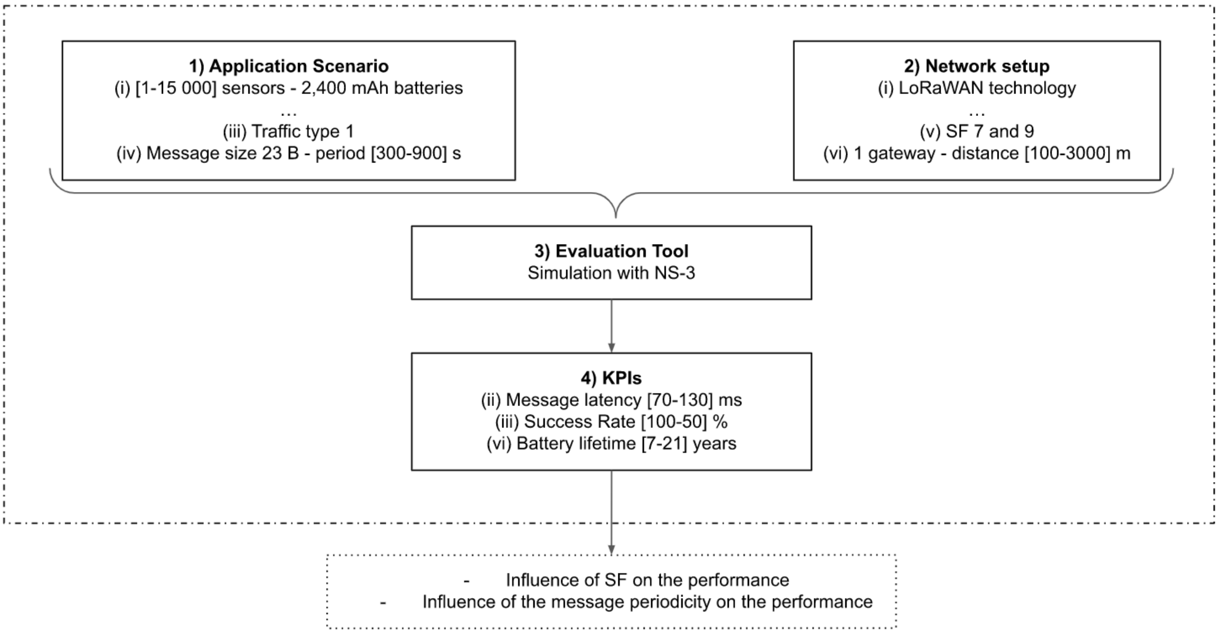


Figure 4.4: Illustration of the Evaluation Framework for Case Study A.

As shown by Figure 4.3a, the reliability remains relatively high until a couple of thousands of sensors regardless of the specific configurations in use for the SF and the packet generation periods. More precisely, we observe that the reliability tends to drop faster when the periods between successive packets are short and when the SF is large (i.e., when sparser modulations, which keep the radio channel busy for a longer time, are in use). Note that we did not represent the packet message latency as the latter does not depend on the number of concurrent sensors. Indeed, unlike Wi-Fi, LoRaWAN does not belong to the listen-before-talking protocols and does not include packet retransmissions so the message latency is not workload-dependent. For an SF of 7 and 9 and a packet of 23 bytes, the message latency is equal to 72 and 230 ms, respectively. These values are typically compliant with the performance requirements of telemetry systems. Figure 4.3b represents the energy efficiency ratio for LoRaWAN in our example of telemetry for a reliability larger than 50%. We notice that the energy efficiency ratio tends to deteriorate with the number of concurrent sensors due to the increasing probability of collisions that reduce the number of bytes successfully conveyed. We can also note the Spreading Factor has a stronger impact on this metric than the period. The use of a small SF is more energy efficient than using a higher SF.

Table 4.6 reports the energy consumption by each sensor over the 3600 seconds of simulation as well as the corresponding lifetime of their battery given their capacity. As expected, we observe that using a smaller SF (i.e., more robust modulation) for the packet transmission results in a longer battery lifetime. The table also shows that depending on the selected period between packet generations, battery lifetime may range from a decade to several tens of years.

We conclude this case study by observing that the scalability index is

Period (s) \ SF	Energy consumption (mJ) 7	Energy consumption (mJ) 9	Battery lifetime (Years) 7	Battery lifetime (Years) 9
300	250	442	13	7
600	134	229	25	14
900	94	158	36	21

Table 4.6: Case study A: Energy consumption per device over 3600 seconds and battery lifetime for LoRaWAN simulations.

more impacted by the Spreading Factor than by the packet generation period. Still, a LoRaWAN based telemetry system can handle between 3,000 and 15,000 stations.

4.4.2 Case study B: Telemetry on Wi-Fi

In our second example, we consider a telemetry system in which Wi-Fi is used to send data from the sensors (end-devices) up to the access point (gateway).

For the application scenario, the sensors are located in the vicinity of the gateway (deployment scope of 20 meters), and their number can vary from 1 to 60. We consider that their batteries have a capacity of 5,200 mAh powered by 12 V. The traffic originating from the sensors also matches type 1 of Table 4.1. For the workload, we assume two possible sizes for the packets: 23 and 1,000 bytes as well as 3 possible periods for the rate at which packets are generated by each sensor: 6, 60, and 360 seconds. We also assume a sub-urban radio environment (path loss model to represent the radio propagation model). The application modeling of this case study can be found in Table 4.7. For the network setup, we consider that cameras will use the 802.11ac amendment of the IEEE 802.11 standard on the 5 GHz with a channel width of 80MHz, a single spatial stream, without frame aggregation and long guard intervals. We set the MCS value to a value of 9. The network also has a star topology with one gateway. We also use ns-3 to evaluate the performance of the considered network. Performance parameters such as the attained throughput, message latency, and reliability are rather straightforward to obtain from the simulator execution.

To estimate the energy cost of communications, we use the ns-3 module that was developed based on the energy model of Wu et al. [156]. The power consumption of Wi-Fi communications is also estimated thanks to a state machine, which maps values of drawn current to each possible state of the Wi-Fi NIC. We calibrated the drawn current parameters of each state using the experiments provided by Serrano et al. in [130]. The associated drawn currents are provided in Table 4.8. Through this model, which is embedded within ns-3, we are able to compute the power consumption of any sensor resulting from its Wi-Fi communications. We can also easily obtain the energy efficiency ratio as well as the expected battery lifetime.

[156]: Wu et al. (2012), 'An Energy Framework for the Network Simulator 3 (ns-3)'

[130]: Serrano et al. (2014), 'Per-frame Energy Consumption in 802.11 Devices and its Implication on Modeling and Design'

Table 4.7: Application Modeling of Case B.

Application modeling	Parameters	Case B
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	1 60 5.2
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream {23,1000} 0.002 0.16
Environment	• Type • Scope (meters) • Expected lifetime (days)	Suburban 20 N/A

State	Drawn current value (mA)
Tx	107
Rx	40
CCA Busy	1
Idle	1

Table 4.8: Drawn current values for each state of the machine state used in ns-3 simulations to evaluate the power consumption of Wi-Fi communications.

For a total number of sensors between 1 and 60, the simulation results show that the reliability for the packet transmission is kept to 100% and that the message latency remains at its lowest level (below tens of milliseconds and then much lower than the typical requirements for telemetry). These results owe to a total workload having its maximal value at 0.07 Mbps (with 60 sensors, packet size of 1,000 bytes and a periodicity of 6 seconds) when the radio channel supports a data rate of 50 Mbps. The power consumed by Wi-Fi for each sensor (due to the exchange of communication with the access point) amounts to nearly 47 J for a simulation time of 3600 sec regardless of the number of concurrent sensors. Interestingly, this value also remains about the same for the different combinations of packet sizes and periodicity. This underlines the important energy overhead brought by Wi-Fi resulting from the lack of sleep state in most 802.11 implementations (unlike LoRaWAN) and, to a lesser extent, from the frequent receptions of beacons sent by the access point every 100 ms. Having computed the consumed power resulting from the Wi-Fi communications, we can derive the expected battery lifetime, which we find to be approximately 200 days.

Given that we arguably tested what could be the upper bounds for the packet size and periodicity for the purpose of telemetry applications, we can conclude that Wi-Fi will do network-wise (provided that its radio range is enough) but that the batteries of sensors will typically last less than a year unless they have some form of self-harvesting capabilities or are on electric supply.

Then, we compute the energy efficiency of Wi-Fi using packets of larger sizes. Table 4.9 reports the energy efficiency ratios obtained for a packet size of 23 bytes as well as those measured when the packet size is set to 1,000 bytes for three different rates of packet generation (i.e., periods of 6, 60 and 360 seconds). As expected, we observe that the energy efficiency ratio grows significantly with the size of packets (and decreases with the period between packet generations). Regardless of the considered period for the time between packet generation, increasing the packet size by 43 fold (from 23 to 1,000 bytes) approximately results in a 30-fold increase of the energy efficiency ratio.

Interestingly, we observe that the values obtained here are worse than those obtained for LoRaWAN with a packet size of 23 bytes but that they somehow reach the same efficiency as Wi-Fi if the latter uses packets of size 1,000 bytes with a period of 60 s.

Packet size (Bytes) \ Period (sec)	6	60	360
23	0.44	0.04	0.007
1,000	13.67	1.36	0.21

Table 4.9: Case study B: Energy efficiency ratio (kBytes/Joule) with Wi-Fi for different sizes of packets and packet generation rates.

4.4.3 Case study C: Video-surveillance on Wi-Fi

In this example, we study how Wi-Fi can be used to support the communication exchanges in a video-surveillance application. Such a scenario is expected to be strongly supported by IoT networks [157]. Each camera (end-device) generates a stream of video frames that are sent upwards to the access point (gateway) and then transferred to a back-end server. We do not explore this use case on LoRaWAN as the minimum bandwidth requirement for video traffic (a minimum of 1Mbps) is not met with a low power technology like LoRaWAN (providing a maximum of 27kbps).

[157]: Rego et al. (2018), 'An Intelligent System for Video Surveillance in IoT Environments'

The application scenario is defined as the following: Cameras are located in the proximity of the gateway and their number varies from 1 to 60, with the same batteries as for Case Study B and also with UDP as the transport protocol. Note that this traffic corresponds to type 3 from Table 4.1. For the workload, we consider three different rates for the application rate: 2, 5, and 8 Mbps which can be viewed as three different codecs, corresponding to real video traces [158] having frames of different sizes, with a fixed FPS (Frames Per Second) of 30. The application modeling of this case study can be found in Table 4.10. The same network setup parameters are used as for Case Study B, with the only difference that the MCS takes a value either of 6 or of 9. The former MCS represents a medium value for the data rate of the radio channel while the latter represents a high value. Simulation with ns-3 is also used as the evaluation tool.

We now turn to the simulation results summarized in Figure 4.5. First, looking at Figure 4.5a, we observe that Wi-Fi can sustain up to 8 or 9 cameras when each of them generates a stream of 8 Mbps. Beyond 9 cameras, the reliability rapidly decreases with packets being dropped as the radio channel activity increases. Using a lower video codec like 5 Mbps and 2 Mbps allows to expand the maximum number of supported cameras to 15 and 30, respectively. Interestingly, we notice that the value of MCS does not impact the results here. As expected, the message latency increases with the number of cameras connected to the access point (see Figure 4.5b). Although it rapidly increases with the number of cameras, its absolute value remains relatively low and does not affect the good behavior of the system even for a total of 40 cameras. Because a video-surveillance system with a reliability below 60% can be considered as a non-functional system, we limit our analysis in Figures 4.5c and 4.5d to cases where the number of cameras leads to a reliability larger than 60%. Figure 4.5c indicates that the number of successfully delivered bytes per joule over Wi-Fi mostly depends on the number of concurrent

Table 4.10: Application Modeling of Case C.

Application modeling	Parameters	Case C
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	1 60 5.2
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream [1000,5000] 30 FPS 30 FPS
Environment	• Type • Scope (meters) • Expected lifetime (days)	Suburban 20 N/A

cameras as its value can decrease 10-fold, ranging from a bit more than 15 MBytes per joule when there is only one camera with MCS 9 and a video rate of 8 Mbps up to less than 3 MBytes per joule for a total of 29 cameras with MCS 6 and a video rate of 2 Mbps. Finally, Figure 4.5d represents the estimated lifetime of the battery. The results demonstrate the importance of having a low video rate to improve the battery lifetime especially if the number of cameras remains relatively low, say no more than 10.

Overall, we observe with this case study that the scalability index (the maximum number of cameras that can be connected to a gateway without degrading the performance) strongly depends on the rate of the video data stream and much less on the selected MCS. A Wi-Fi based video-surveillance can handle between 5 and 20 cameras, which can live on their battery for a month or two [159] depending on the used codecs.

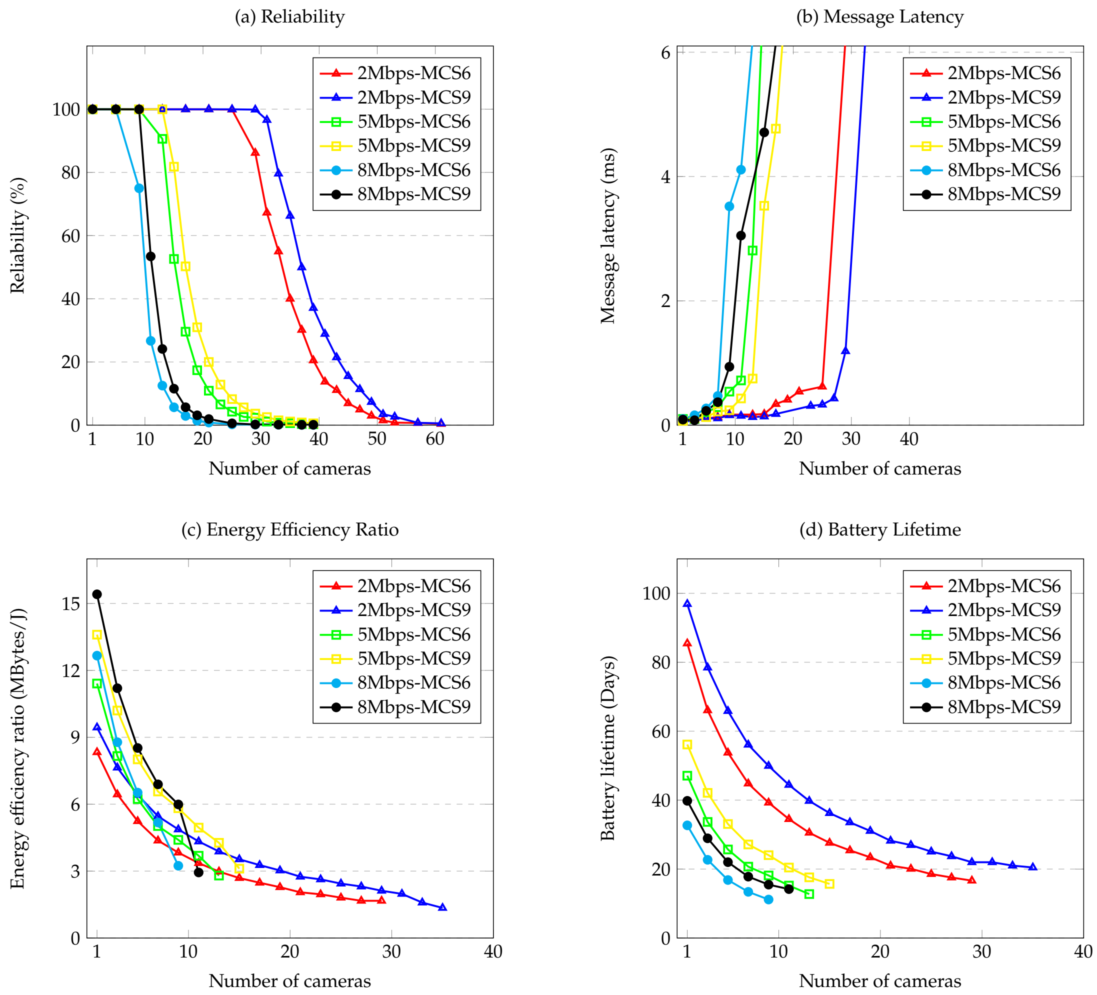


Figure 4.5: Case study C: KPIs for Video-surveillance on Wi-Fi.

4.5 Discussion

The results of the previous section regarding telemetry, have confirmed the superiority of LoRaWAN setup in terms of scalability. Up to thousands of sensors can be managed by a single gateway. Our results show that, despite being almost an order of magnitude more energy-efficient (in terms of Bytes successfully transmitted per joule) than LoRaWAN when the end-devices have a lot of data to exchange, Wi-Fi is significantly overpowered by LoRaWAN for the battery life of their end-devices (not mentioning its shorter radio range) in the case of a telemetry application. Then the key guidelines for deploying a train passengers metering system using LoRaWAN would be:

- If the distance between the gateway and sensors is not very large, lower than a thousand meters, then privileging lower SFs will ensure more reliability and less energy consumption. This will be for example the case for the train passengers metering solution.
- Message periodicity should be carefully set since it may strongly influence the performance and longevity of the system. Charts provided by the framework will be used to guide the decision.

Note that for both LoRaWAN and Wi-Fi, we can consider other kinds of radio channels (more noisy for example) by using different propagation models [160]. This would be another round of experiments that the architect would run to refine its configuration with respect to the environmental context.

[160]: Benin et al. (2012), 'Vehicular Network Simulation Propagation Loss Model Parameter Standardization in ns-3 and Beyond'

On an other side, an IoT architect looking for a network setup for a video-surveillance application may select Wi-Fi for its strong reliability. The following guidelines can be generated from the previous evaluation results:

- One Wi-Fi access point can manage at least ten cameras.
- The greater the MCS, the better the performance will be.
- Favoring low mean data rates for the video streaming may strongly influence the scalability and lifetime of the system, even though the quality of the video may suffer from it.
- The architect will have to use the results to derive the best compromise between the required video quality and battery lifetime.

4.6 Conclusion

We have presented a framework to evaluate the performance of the network technologies for IoT applications where multiple end-devices exchange data via gateways. The framework includes the definition of a scenario and of its KPIs as well as their evaluation. We used two typical use cases, inspired by real-life IoT applications, to illustrate the applicability of our framework on different network technologies. At this stage, we paid special attention to the energy efficiency as well as to the ability of an IoT communication technology to properly scale up with the number of end-devices. The provided application-based evaluation results highlight the importance of having a holistic approach when evaluating the good fit of a network technology in its field context. It allows: (i) Users and researchers to rapidly evaluate the applicability of a network technology to an IoT scenario and (ii) network technology or service providers to present the effective KPIs of their product or service in a standardized framework to accelerate comparison, qualification and selection. The code repositories for our numerical results are available in [161] and [162].

Regarding the limitations of this work, we can quote the following: First, it considers a simple topology with a single gateway for each network technology. This may reduce the scope of the possible IoT solutions to be applied for. Also, it lacks an automatic selection process in case we have competing candidates for a given network application (LoRaWAN or Wi-Fi HaLow for instance).

We address these limitations in the next chapter. We extend our framework using decision-making tools to automate the comparison of multiple technologies (LoRaWAN, Wi-Fi HaLow, 802.15.4, 5G) and to select the best alternative among them. We also analyze the impact of the topology of the network in terms of gateways, and the trade-off between the yielded performance improvement and the cost.

IoT Network Technologies Selection 5

5.1 Introduction

We have presented in the last chapter a framework for modeling IoT applications and network setups and for assessing the performance of a setting for a given IoT application and its workload. Now, there is still the need for a decision-support methodology and an associated tool to systematize the evaluation and comparison process. This would enable to objectively compare technology candidates and to deliver KPI to support decision-making at the IoT solution's design and ongoing management phases. The ultimate goal is to future-proof and select the most appropriate network technology for new application development or an adequate configuration for a new deployment of a pre-packaged IoT solution. However, finding the right abstractions that lead to a good balance between performance accuracy and computational complexity when comparing networking options remains an open and challenging issue.

Selecting the best network technology alongside its topology (number of gateways) and configuration is critical for the success of an IoT solution. Recent studies (e.g., [163, 164]) have shown that almost 75% of IoT projects, be it in the US, UK, or India, were deemed failures and that 30% of IoT projects failed to move beyond the proof-of-concept stage. While customers see real value in deploying IoT, many industrial companies and projects are lagging – for example up to 70 percent of industrial companies' projects end up in "pilot purgatory" [165]. The skills shortage and the difficulties in navigating the technological ecosystem are part of the barriers that explain these failures. The profusion of possibilities often results in non-decision, non-optimal choices, the excessive total cost of ownership and, ultimately, project failure. For instance, the success of an IoT project can be jeopardized due to an insufficient budget or wrong technological decisions such as an inadequate network technology.

[163]: Beecham Research (2020), 'Why IoT projects fail & how to beat the odds An executive summary'
[164]: Microsoft (2022), 'IoT Signals - Manufacturing Spotlight'

[165]: Aramayo-Prudencio et al. (2018), 'Digital Manufacturing–Escaping Pilot Purgatory'

In this chapter, we focus on the performance and scalability of IoT network technologies that critically impact the ability of the IoT solution to support the required profile and to handle increased processing volumes in the future if required. We formalize and investigate the design optimization problem for selecting and configuring the IoT network technology of an application that can evolve over time. We leverage a fine balance between performance accuracy and computational complexity to provide a methodology, namely HINTS (metHodology for IoT Network Technology Selection), that combines IoT application requirements, goals formulation, IoT network modeling, discrete-event network simulation, and a Multi Attibute Decision Making (MADM) method. The main contribution of this chapter is to propose a formal approach and associated algorithms to automatically optimize the selection of an IoT network technology for a given application.

The remainder of the chapter is as follows: In Section 5.2 we formulate the problem of IoT network technology design and configuration. In

Section 5.3, we present our solution to the problem. Section 5.4 is devoted to the illustration of the implementation of the methodology on three use-cases. Finally, Section 5.5 concludes this chapter.

5.2 Problem Formulation

5.2.1 General Description

We formulate this design optimization problem for IoT network performance and scalability that we are addressing as follows. Let us consider:

- An IoT application A, with its set R of characteristics and communication requirements.
- A set T of network technologies candidates.
- For each network technology T_i in T, a set C_i of possible network configurations.
- A set K of key performance metrics or KPIs that characterize the behavior of an application A on a network technology T_i with C_i.
- A set G of performance goals, defined as thresholds targeted by the application designer for each KPI.

The decision problem consists in finding the network technology T_d in T and the associated network configuration C_d^k (k-th configuration of the network technology T_d) that fit the application requirements R and best match the performance goals G of the application A, in terms of KPI K.

The decision problems are formulated in detail in Problem Formulation 1. Note that we categorize the KPIs in two classes: (i) Without threshold and (ii) with a threshold. The former class comprises KPIs for which the goal is defined as "the more (or the less), the better". This can be the case of the cost KPI for example. The second class includes KPIs for which a goal can be quantified as a threshold. This means that there is no need to go beyond (or below) a given threshold, as further discussed in Section 5.3.6.

Also, the focus in this chapter is on network technologies based on a star topology. Mesh networks as well as hierarchical network interconnections leveraging routing protocols such as RPL [166], represent alternative architectures for IoT connectivity. They raise interesting additional decision questions such as determining the most power-efficient routing and load-balancing strategies that are not considered in this work.

[166]: Winter et al. (2012), *RPL: IPv6 Routing Protocol for Low-power and Lossy Networks*

Problem Formulation 1: Problem Formulation for the IoT Network Technology Selection Problem

1: **Inputs**:

- **Application**:
  $R = [R_1, \ldots, R_k]$; Application requirements (*e.g.*, min and max number of devices, size and frequency of messages, etc.).
- **KPIs**:
  $K = [K_1, \ldots, K_n]$, KPIs (*e.g.*, Reliability, Battery lifetime, Message latency, Cost, etc.).
- **KPIs performance goals**:
  $G = [G_1, \ldots, G_n]$, $G_i \in \mathbb{R}_+$; KPIs performance goals (or KPI thresholds) ($G_i < 0$ means that the i-th KPI is without threshold).
- **KPIs weights**:
  $W = [W_1, \ldots, W_n]$, $W_i \in \mathbb{R}_+$, $\sum_{i=1}^{n} W_i = 1$; Weights attributed to each selected KPI.
- **Set of network technologies candidates**:
  $T = [T_1, \ldots, T_m]$; IoT network technologies (*e.g.*, LoRaWAN, LTE-M, Wi-Fi, etc.).
- **Network configuration parameters**:
  $C = [C_1, \ldots, C_m]$; Network configurations per network technology (*e.g.*, SF for LoRaWAN, MCS for Wi-Fi, etc.).

2: **Outputs**:

- **Decision D1**: Select the network technology T_d that best matches the KPIs goals.
- **Decision D2**: Select the network configuration C_d^k for the chosen network technology T_d which best matches the KPIs goals.
- **Decision D3**: Select the minimal number of gateways g_d for the chosen network technology T_d which best matches the KPIs goals.

5.3 HINTS Description

5.3.1 Overview of the Methodology

The HINTS methodology targets simplicity, efficiency and risk limitation to address the IoT network decision problem formulated in Algorithm 1. To attain these objectives, the HINTS methodology is divided into two parts: (i) The network modeling part, which addresses the concern related to network experts and (ii) the application-driven decision part, which addresses the needs of application architects.

The network technology modeling part consists in abstracting and quantifying the relevant parameters of network technologies. We use the models provided in Section 4.3.2 of Chapter 4.

The application-driven decision part of HINTS, illustrated by Figure 5.1, is divided into 5 steps as follows:

1. **Application modeling**, where the value of the application requirements, listed in Section 4.3.1, KPIs performance goals and weights are defined.

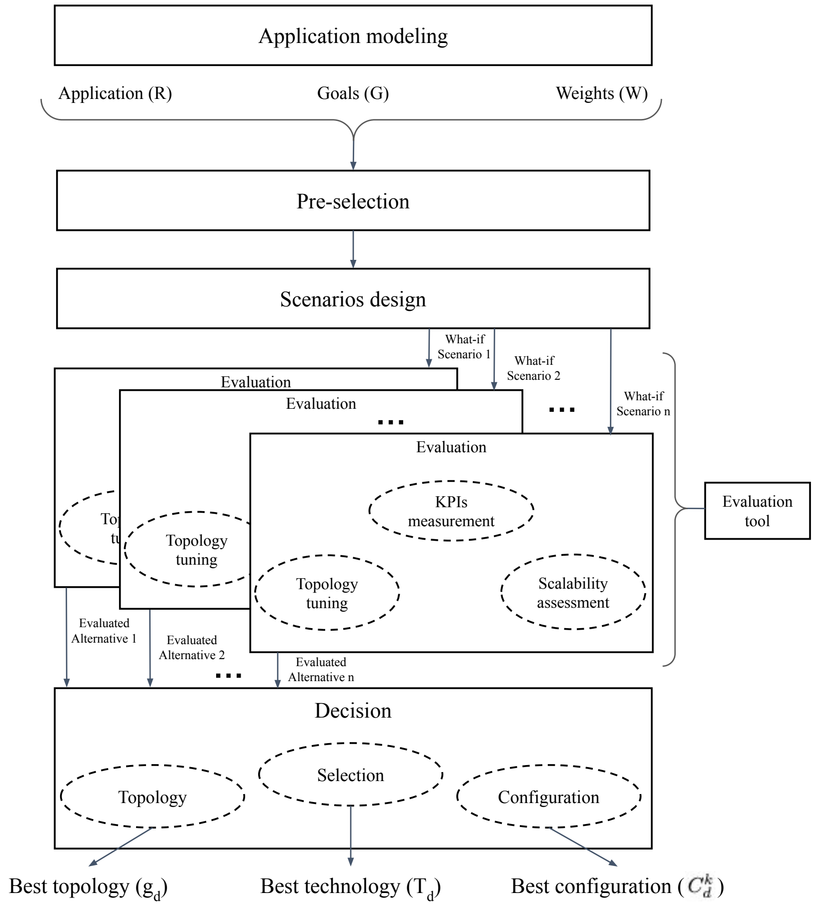


Figure 5.1: Overview of the HINTS Steps.

2. **Pre-selection**, where network technologies candidates are filtered based on their technical specifications and on the application requirements.
3. **Scenario design**, where what-if scenarios, integrating the application with remaining network technologies, are designed.
4. **Evaluation**, where the what-if scenarios are instantiated and executed on an evaluation environment and the KPIs of each what-if scenario are obtained.
5. **Decision**, where what-if scenarios are ranked and the best network technology and its associated configuration are identified via a multi-criteria decision-making approach.

5.3.2 Application Modeling

The **application modeling** step aims at:

- Quantifying the application requirements (an example for a smart building use-case is given in Table 5.1).
- Specifying the targeted KPIs and defining their performance goals.
- Attributing weights to KPIs.

An example of the KPIs, thresholds and weights for a smart building use-case is provided in Table 5.2. Recall that $G_i < 0$ means that the i-th KPI is without threshold.

Table 5.1: Example of a Smart Building Application Requirements.

Application abstraction parts	Parameters	Value
End-devices	Minimal number	50
	Maximal number	100
	Battery capacity (Amperes.hour)	2.4
Workload	Traffic direction	Upstream
	Message size (bytes)	100
	Minimal frequency (packets/second)	1
	Maximal frequency (packets/second)	1
Environment	Type	Indoor
	Scope (meters)	100
	Expected lifetime (days)	730

Table 5.2: Example of KPIs Goals and Weights.

KPI name	Unit	Goal	Weight
Reliability	Percentage (%)	90	0.25
Battery lifetime	Days (d)	60	0.25
Message latency	Miliseconds (ms)	10	0.25
Cost	Dollars ($)	-1	0.25

5.3.3 Pre-selection

The goal of the **pre-selection** step is to dismiss network technologies that are obviously not meeting the application requirements. For example, a network technology can be dismissed because its maximum data rate does not support the expected workload, derived from the message frequency and the message size. To do this, HINTS applies a filtering process, which can be implemented as a decision tree. The application requirements are compared to the maximum values of the message size and data rate that a network technology can provide. For instance, an application scenario with a traffic workload over 1 Mbps can never be satisfied with LoRaWAN. Then, there is no need for further analysis. The inappropriate network technologies are simply dismissed for the following steps.

5.3.4 Scenario Design

After the **pre-selection** step, there is a need to explore in depth the network technologies candidates with various network configurations in order to compare them. The **scenario design** step consists in identifying the different network settings for the network technology candidates. Each setting represents a network technology candidate associated with network configuration parameters to be evaluated and compared to the others. This means that for a single network technology, there can be various network configurations where each one represents a what-if scenario (for instance, LoRaWAN with SF7 will be considered differently than LoRaWAN with SF12). Note that most of the considered network configuration parameters are naturally bounded (*e.g.*, SF, from 7 to 12).

Scalability and evolutivity assessment

Most IoT deployments are expected to evolve over time, for instance in terms of network density (number of end-devices) or in terms of traffic workload (message frequency and message size). The future behavior of a network technology under these conditions must also be evaluated. HINTS recommends designing scenarios with the maximum number of end-devices and the heaviest traffic workloads. To this end, every what-if scenario is composed of a minimal deployment (with the minimal number of end-devices and the minimal message frequency), and a maximal deployment (with the maximal number of end-devices and the maximal message frequency). Recall that these parameters have been defined in Section 4.3.1. This will provide insights into the scalability and evolutivity of the different what-if scenarios.

5.3.5 Evaluation

The **evaluation** step consists in instantiating the what-if scenarios defined at the **scenario design** step described above for calculating their respective KPI values, for both the minimal and the maximal deployments. For the evaluation, we rely on the framework presented in Chapter 4. We categorize KPIs in two classes: (i) Without threshold and (ii) with a threshold. The former class comprises KPIs for which the goal is defined as "the more (or the less), the better". This can be the case of the cost KPI for example. The second class includes KPIs for which a goal can be quantified as a threshold. This means that there is no need to go beyond (or below) a given threshold, as further discussed in Section 5.3.6.

Here also, real experimentation or simulation tools can be used as evaluation tools. If experimentation is used, end-devices and monitoring tools must be set up and activated to capture the traces. Then the traces have to be analyzed and the KPIs computed. For small-scale projects and a limited number of technologies and scenarios, this can be done in labs and within a reasonable time. If the number of scenarios or end-devices is important, the experimentation may be impossible to perform. Moreover, experimenting with a variety of technologies requires rare talents in all these network technologies. Simulation can be considered as a better approach to get a decent evaluation of the KPIs. Even if the simulation can be viewed as providing approximate results, it allows relative scalability and reconfigurability, which are required to explore different configurations for each network technology.

Topological considerations

As an extension to our evaluation framework, we propose to study here the impact of the topology of the IoT network. Indeed, for a given network technology with its configuration, the number of gateways deployed, g, referred to as the topology throughout this chapter, can greatly impact the application performance. Increasing the parameter g may have three main implications on the network behavior and the performance of the communications:

1. Reducing the workload per gateway: We assume that each gateway has a dedicated channel so that the workload at each gateway decreases proportionally with the total number of gateways. This assumption is realistic for many technologies that have multiple orthogonal channels (*e.g.*, 64 in 868 MHz for LoRaWAN, 24 in 5 GHz for Wi-Fi).
2. Reducing the maximum distance between end-devices and their associated gateway: In our application model, we consider that the maximum distance between an end-device and its associated gateway is:
$$d = D/(2 * g) \tag{5.1}$$
where D is the deployment scope (the maximum distance between two end-devices, see Sec. 4.3.1). This simple relation reflects that, in general, the more gateways, the closer the end-devices are from their gateway.
3. Increasing the cost of the solution: It can incur additional costs in the purchase, but also in the deployment and the maintenance of the gateways.

Therefore, the ideal topology (ideal number of gateways) has to be determined for each what-if scenario. The HINTS approach is to evaluate each what-if scenario with an increasing number of gateways, g, starting at the minimal number (typically 1). The parameter g is iteratively increased by 1 until either KPIs goals of all the threshold-based KPIs (namely, reliability, battery lifetime and message latency) are reached (depending on the defined goals G), or the improvement on these (threshold-based) KPIs is below a given value ϵ. To include a safety margin, the upper bound on g is incremented by one. Overall, the number of explored topologies for each what-if scenario is simply equal to the maximal number of gateways that were iteratively tested. This process is described in Algorithm 1. Throughout the chapter, we define an alternative as a what-if scenario associated with a topology.

HINTS proposes the following formula to compute the cost KPI, including the deployment (network modules of the end-devices, and gateways) and the maintenance costs (*i.e.,* battery change):

$$Cost = \underbrace{p_{gw} * n_{gw} + p_{ed} * n_{ed}}_{Deployment} + \underbrace{(l/b) * p_{br} * n_{ed}}_{Maintenance} \tag{5.2}$$

where each parameter is defined in Table 5.3. The ratio of l (expected application lifetime) on b is used to calculate the number of times the end-devices' batteries will have to be replaced. Note that b (battery lifetime) is the only KPI whose value is derived from the simulation and not obtained directly.

5.3.6 Decision

The goal of the **decision** step is to compare and rank the alternatives evaluated in the **evaluation** step. The KPI values obtained in the **evalu-**

Table 5.3: Cost Function Parameters.

Symbol	Signification
p_{gw}	Price of a gateway
n_{gw}	Number of gateways
p_{ed}	Price of a network module for the end-device
n_{ed}	Number of end-devices
l	Expected scenario lifetime
b	Battery lifetime
p_{br}	Cost of a battery replacement

ation step are stored in a matrix P. Depending on the class of the KPI (see Section 5.2), its original value is kept or it is caped (or floored) by a threshold, as shown in Algorithm 2. For the second class (threshold-based KPIs), in the case of a minimum threshold (*e.g.*, for reliability or battery lifetime), we have:

$$f(x,\alpha) = \begin{cases} x & \text{if } x > \alpha \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

and for a maximum threshold (*e.g.*, message latency), we have:

$$f(x,\alpha) = \begin{cases} \alpha & \text{if } x < \alpha \\ x & \text{otherwise} \end{cases} \tag{5.4}$$

where x denotes a KPI and α its associated threshold.

Then, the KPIs values are normalized as shown in Algorithm 3:

[137]: Tzeng et al. (2011), *Multiple Attribute Decision Making: Methods and Applications*

The results are ranked according to a score, obtained through a method derived from the TOPSIS MADM algorithm [137]. In HINTS, the ranking leverages (i) KPIs weights and (ii) a scalability factor set by IoT architects, on the basis of their knowledge of the business context. The KPIs weighting is done using a vector of preference, more commonly named weights, in the form of $W = [W_1, \ldots, W_n]$ where $W_j \in \mathbb{R}$, $\sum_{j=1}^{n} W_j = 1$. The scalability factor, $\beta \in \{0,1,2\}$, determines which one of the minimal or the maximal deployment has more importance for decision making ($\beta = 0$ or $\beta = 2$, respectively), or if they have the same importance ($\beta = 1$) according to the IoT architect. If the scalability of the solution in terms of the number of end-devices or in workload intensity (maximal number of end-devices and maximal message frequency, respectively) is critical, the architect will give a high value, namely 2, to the scalability factor. The weighted KPIs will then be multiplied by this scalability factor for the "at scale" (aka the maximal deployment) evaluation of a scenario. This process is detailed in Algorithm 4. Note that, for simplicity purposes, we consider that the obtained KPIs of alternatives are organized as follows: The first n KPI values correspond to the minimal deployment, whereas the n remaining KPI values correspond to the maximal deployment, as shown in Equation 5.5.

$$p_i = [\underbrace{p_{i1}, \ldots, p_{in}}_{Min.deployment}, \underbrace{p_{i(n+1)}, \ldots, p_{i2n}}_{Max.deployment}] \tag{5.5}$$

HINTS calculates the positive ideal solution (best one) and the negative ideal solution (worst one) based on the range of estimated KPIs values.

Algorithm 1: Evaluation Process

Inputs:
$R = [R_1, \ldots, R_k]$; Application requirements;
$T = [T_1, \ldots, T_m]$; Pre-selected IoT network technologies;
$C = [C_1, \ldots, C_m]$; Network configurations;
$G = [G_1, \ldots, G_n]$, $G_i \in \mathbb{R}$; The KPIs performance goals (either a KPI threshold, or $G_i < 0$ means that the i-th KPI is without threshold);
$\epsilon \in \mathbb{R}$; Minimum improvement for KPIs;
Variables:
$q \in \mathbb{N}$; Number of evaluated alternatives;
$P = (p_{ij}) \in \mathbb{R}^{q \times 2n}$; KPI values of the alternatives;
Algorithm:
$ind = 1$;
for each i in $[1;m]$ **do**
  $g \leftarrow 1$; $limit[i] \leftarrow \infty$; $search \leftarrow True$;
  $p_0 \leftarrow \underbrace{[0, \ldots, 0]}_{n}$
  while $g \leq limit[i]$ **do**
    for each j in $[1;|C^i|]$ **do**
      $p_{ind} \leftarrow$ Evaluation (R, C_i^j, g) // Evaluation returns the KPI values for the network configuration C_i^j, with g gateways.
      if (KPIs_Satisfied (p_{ind}, G) or Improvement $(p_{ind}, p_{ind-1}) \leq \epsilon$ and $search = True$ **then**
        $limit[i] \leftarrow g + 1$
        $search \leftarrow False$
      end if
      $ind \leftarrow ind + 1$;
    end for
    $g \leftarrow g + 1$
  end while
end for
$q \leftarrow ind$

Algorithm 2: Applying Filters to KPIs

Inputs:
$q \in \mathbb{N}$; Number of evaluated alternatives;
$G = [G_1, \ldots, G_n]$, $G_i \in \mathbb{R}$; The KPI performance goals ($G_i < 0$ means that the i-th KPI is without threshold);
$P = (p_{ij}) \in \mathbb{R}^{q \times 2n}$; KPI values of the alternatives;
Algorithm:
/* Applying KPI thresholds */
for each i in $[1, q]$ **do**
  for each j in $[1, 2n]$ **do**
    if $G_j \geq 0$ **then**
      $p_{ij} \leftarrow f(p_{ij}, G_j)$ // KPI with threshold (see Eqs. 5.3 & 5.4)
    end if
  end for
end for

Algorithm 3: Normalization Process

Inputs:
$P = (p_{ij}) \in \mathbb{R}^{q \times 2n}$; KPI values of the alternatives;
Variables:
$N = (n_{ij}) \in \mathbb{R}^{q \times 2n}$; Normalized KPIs;
Algorithm:
/* Normalization */
for each i in $[1, q]$ **do**
  for each j in $[1, 2n]$ **do**
    $n_{ij} \leftarrow \dfrac{p_{ij}}{\sqrt{\sum_{i=1}^{q} (p_{ij})^2}}$
  end for
end for

Algorithm 4: Weighting Process

Inputs:
$W = [W_1, \ldots, W_n]$, $W_i \in \mathbb{R}_+$, $\sum_{i=1}^{n} W_i = 1$; KPIs weights;
$N = (n_{ij}) \in \mathbb{R}^{q \times 2n}$; Normalized KPIs;
$\beta \in \{0, 1, 2\}$; Scalability factor;
Variables:
$V = (\mathrm{v}_{ij}) \in \mathbb{R}^{q \times 2n}$; Weighted normalized KPIs;
Algorithm:
/* Weighting */
for each i in $[1, q]$ **do**
  for each j in $[1, 2n]$ **do**
    $v_{ij} \leftarrow W_j \times n_{ij}$
  end for
  /* Apply the scalability factor to the KPIs obtained for the maximal deployment*/
  for each j in $[n + 1, 2n]$ **do**
    $v_{ij} \leftarrow \beta \times v_{ij}$
  end for
end for

Then, a score is given to each alternative depending on the Euclidean distances between the considered alternative and the positive and negative ideal solutions. The way of calculating the positive and the negative ideal solutions as well as the scores is described in Algorithm 5, and depicted in Figure 5.2.

Finally, the output of the **decision** step is the alternative that obtains the highest score, according to this ranking.

5.3.7 Summary of the Methodology

Using HINTS, IoT architects are able to leverage the network knowledge previously encoded by network experts, and make wise decisions by:

1. Quantifying the application requirements, identifying the KPIs performance goals and weighting them in the **application modeling** step.
2. Quantifying the KPIs performance goals to allow the assessment of the network configuration to the specific application context and

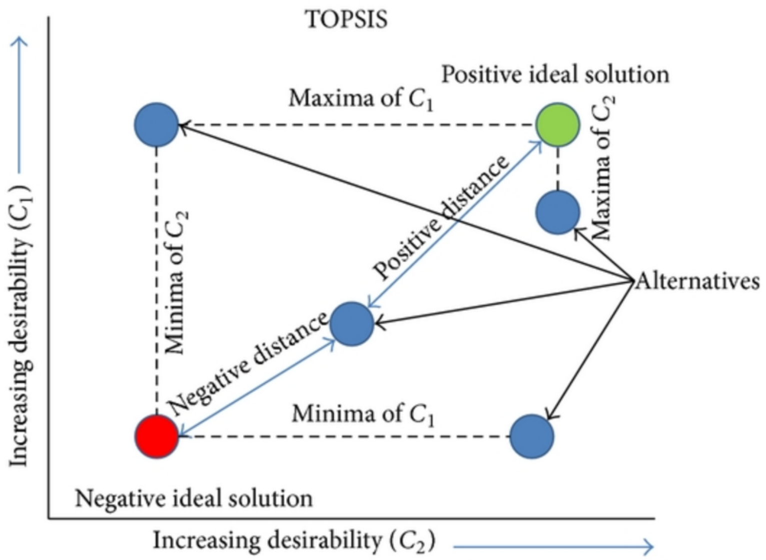


Figure 5.2: Graphical Representation of TOPSIS with Two Attributes [167].

its potential scale in the future in terms of number of devices as well as message frequency, still in the **application modeling** step.

3. Dismissing network technologies that are obviously inappropriate (do not meet the application requirements) via the **pre-selection** step.
4. Specifying detailed what-if scenarios (with network configurations) for the remaining network technologies candidates for in-depth performance and scalability analysis with the **scenario design** step.
5. Evaluating these what-if scenarios with different topologies for the minimal and the maximal number of end-devices and message frequency, in the **evaluation** step.
6. Comparing the alternatives (with network configurations and topologies) and selecting the best one with the **decision** step.

Algorithm 5: Ranking Process

Inputs: $V = (\mathrm{v}_{ij}) \in \mathbb{R}^{q\times 2n}$; Weighted normalized KPIs
Variables:
$V^{+} = [v_1^{+}, \ldots, v_{2n}^{+}]$, $v_i^{+} \in \mathbb{R}$; Ideal positive solution;
$V^{-} = [v_1^{-}, \ldots, v_{2n}^{-}]$, $v_i^{-} \in \mathbb{R}$; Ideal negative solution;
$S^{+} = [s_1^{+}, \ldots, s_q^{+}]$, $s_i^{+} \in \mathbb{R}$; Positive distances;
$S^{-} = [s_1^{-}, \ldots, s_q^{-}]$, $s_i^{-} \in \mathbb{R}$; Negative distances;
Algorithm:
/* Ranking */
for each j in $[1, 2n]$ **do**
  $v_j^{+} \leftarrow Argmax\{v_{ij}, i = 1, \ldots, q\}$
  $v_j^{-} \leftarrow Argmin\{v_{ij}, i = 1, \ldots, q\}$
end for
for each i in $[1, q]$ **do**
  $s_i^{+} \leftarrow \sqrt{\sum_{j=1}^{2n}(v_j^{+} - v_j^{-})^2}$
  $s_i^{-} \leftarrow \sqrt{\sum_{j=1}^{2n}(v_j^{-} - v_j^{-})^2}$
end for
for each i in $[1, q]$ **do**
  $S_i \leftarrow \dfrac{s_i^{-}}{s_i^{-} + s_i^{+}}$
end for

5.4 Case Studies

In this section, we illustrate the implementation of HINTS methodology and its application on three case studies derived from real-life examples: Smart building, event video-surveillance and precision agriculture. We illustrate how HINTS can be leveraged to support the following decisions and context:

- Case A: Network technology and topology decision at the design phase of a tailored smart building solution with a potentially growing number of end-devices.
- Case B: Network technology and topology decision for the design phase of a pre-packaged event video-surveillance solution with a potentially growing traffic workload.
- Case C: Network configuration decision at the deployment phase of a pre-packaged LoRaWAN-based precision agriculture solution.

The HINTS methodology implementation tool [168] provides the following set of network technologies: (i) LoRaWAN, (ii) Wi-Fi HaLow (aka IEEE 802.11ah on the 868 MHz frequency band), (iii) Wi-Fi (namely, IEEE 802.11ac on the 5 GHz frequency band), (iv) 802.15.4 (6LoWPAN) and (v) Private 5G (mmWave on the 28 GHz frequency band). HINTS defines their network configuration parameters as the ones specified in Table 4.3.

In the HINTS implementation, the **pre-selection** step is based on the maximum data rate and the maximum message size for each considered network technology. Table 5.4 enumerates the different "theoretical" values proposed by HINTS for these parameters.

Table 5.4: Numerical values for the maximal data rate and message size on the subset of network technologies, considered in the HINTS implementation tool.

Features \ Technology	LoRaWAN [75]	Wi-Fi [169]	HaLow [170]	802.15.4 [63]	5G mmWave [95]
Maximum data rate	50 Kbps	3.4 Gbps	234 Mbps	250 Kbps	10 Gbps
Maximum message size	256 B	65535 B	65535 B	65535 B	65535 B

For the **evaluation** step, HINTS implementation uses the release 3.33 of ns-3 for Wi-Fi and 802.15.4 (6LoWPAN) and resorts to code patches not integrated in the official version of Network Simulator 3 but widely used by the research community to evaluate LoRaWAN [155] and Wi-Fi HaLow [171], and the module developed in [172] for 5G mmWave. The length of simulations is determined so that there are at least 200 packets sent per end-device.

[155]: Magrin et al. (2017), 'Performance Evaluation of LoRa Networks in a Smart City Scenario'

[171]: Tian et al. (2016), 'Implementation and Validation of an IEEE 802.11 ah Module for ns-3'

[172]: Mezzavilla et al. (2018), 'End-to-end Simulation of 5G mmWave Networks'

The improvement value, ϵ, to determine the ideal topology is set to 5% (recall that ϵ refers to the improvement threshold in the performance, see Section 5.3.5). The prices of End-device (ED) and Gateway (GW) used to compute the cost of an alternative are reported in Table 5.5[1]. The maintenance cost corresponds to the battery replacement. In HINTS, the price of a battery replacement (parameter p_{bc} in Equation 5.2 and Table 5.5) depends on the application scenario environment. It is set to 5 USD for indoor and urban environments, and 50 USD for rural environments (see Table 5.3). Since the considered network technologies operate on unlicensed frequency bands or private environments, there are no additional band subscription fees.

1: Note that other equipment can be considered for each network technology, therefore leading to other prices as well.

Network technology	ED Price (USD)	GW Price (USD)
LoRaWAN [98]	5	1000
Wi-Fi HaLow [173]	15	1000
Wi-Fi [174]	10	100
802.15.4	30	200
5G mmWave	20	500

Table 5.5: Network equipment price for some network technologies considered for our case studies.

For all our case studies, the end-devices are expected to run on batteries. Therefore we keep only the battery lifetime KPI as it is correlated to the energy consumption.

In the **decision** step, and for the sake of simplicity, uniform weights are used for every KPI, and a scalability factor of 1 is used as well, so that the initial and the maximal deployments have the same importance.

5.4.1 Case A: Network Technology and Topology Decision for a Smart Building Solution

This case study is devoted to the design of a tailored smart building solution, where sensors will collect periodical measurements (room temperature and occupancy sensors, air quality, etc.) to maintain safety and comfort within the facility. The structure of the building is the

Table 5.6: Application Modeling of Case A.

Application modeling	Parameters	Case A
End-devices	• Minimal number	50
	• Maximal number	100
	• Battery capacity (Amperes.hour)	2.4
Workload	• Traffic direction	Upstream
	• Message size (bytes)	100
	• Minimal frequency (packets/second)	1
	• Maximal frequency (packets/second)	1
Environment	• Type	Indoor
	• Scope (meters)	50
	• Expected lifetime (days)	730

following: We consider 20, 10 and 50 meters for its length, width and height, respectively, with 16 floors of 6 rooms in each floor.

In the **application modeling** step of HINTS, the application scenario is defined as follows: We consider 50 and 100 sensors for the minimal and the maximal number of end-devices, respectively, equipped with 2,400 mAh capacity batteries (powered by 3 V). The sensors send 1 packet of 100 bytes every second to their gateway. The maximal message frequency is equal to 1 message per second as well. The environment is indoor since the application operates inside a building. The sensors are randomly placed inside the building around a gateway, with a deployment scope of 100 meters (which is approximately equal to the maximum distance that could separate two points inside that building). For the KPIs goals, this application scenario would require batteries to last at least 3 months, a message latency below 100 ms and a reliability above 90%. For the cost calculation, the parameter l (expected scenario lifetime, see Section 5.3.5) is set to 2 years. Table 5.6 summarizes these parameters.

At the **pre-selection** step, LoRaWAN is dismissed from the list of network technologies candidates since the message frequency required for this case study (1 packet per second) is too high for the maximum data rate of LoRaWAN (see Table 5.4). 5G mmWave is also dismissed since these frequency bands are not expected to be used in this kind of application scenarios, due to their poor penetration capacity [175]. Hence, the remaining network technologies are Wi-Fi, 802.15.4 and Wi-Fi HaLow.

[175]: Jijo et al. (2021), 'A Comprehensive Survey of 5G mm-wave Technology Design Challenges'

At the **scenario design** step, we consider the following network configurations for the remaining network technologies: For Wi-Fi, it is a channel width of 80 MHz, one spatial stream, a long guard interval and no frame aggregation. For Wi-Fi HaLow, it is a channel width of 2 MHz, a long guard interval, a beacon interval of 51200 ms and one RAW group. For 802.15.4, it is a channel width of 5 MHz, a number of frame retries of 4, a number of CSMA backoffs set to 5 and the maximum (resp. minimum) backoff exponent set to 4 (resp. 3). Note that these values are used as a default network configuration, and other parameters can be considered for further study. The same remark applies to the remaining case studies.

Table 5.7: Case A: Smart Building Results.

Network technology		Minimal deployment (50 end-devices)				Maximal deployment (100 end-devices) *Scalability factor: 1*				
Technology	Nb. of GW	**Reliability** *Weight: 1 Unit: % Goal: >90*	**Battery Lifetime** *Weight: 1 Unit: d Goal: >80*	**Message Latency** *Weight: 1 Unit: ms Goal: <100*	**Cost** *Weight: 1 Unit: $*	**Reliability** *Weight: 1 Unit: % Goal: >90*	**Battery Lifetime** *Weight: 1 Unit: d Goal: >80*	**Message Latency** *Weight: 1 Unit: ms Goal: <100*	**Cost** *Weight: 1 Unit: $*	**Score**
Wi-Fi	1	42.0	61.72	0.05	3850	30.0	49.1	0.05	9100	0.02
Wi-Fi	2	80.0	66.28	0.05	3700	86.0	61.24	0.05	7700	0.07
Wi-Fi	3	87.5	66.45	0.05	3800	96.97	85.86	0.05	5800	0.32
Wi-Fi	4	100.0	89.09	0.05	3150	100.0	88.38	0.05	5900	0.46
Wi-Fi	5	100.0	89.27	0.05	3150	100.0	88.71	0.05	5900	0.46
HaLow	1	100.0	362.16	48.41	2250	100.0	277.78	57.28	3500	0.87
HaLow	**2**	**100.0**	**421.69**	**48.9**	**3000**	**100.0**	**331.8**	**58.72**	**4500**	**0.93**
802.15.4	1	54.31	91.76	29.62	3700	44.63	71.44	12.61	9700	0.12
802.15.4	2	94.46	125.07	12.38	3400	88.29	85.75	21.67	7400	0.36
802.15.4	3	98.09	142.95	16.47	3350	94.10	112.28	7.46	7100	0.49

The simulation time is set to 200 seconds in the **evaluation** step. We present the results of the **evaluation** step in Table 5.7. In this step, HINTS iterates to determine the ideal topology. In this example, the KPIs performance goals are met for the threshold-based KPIs with one gateway for Wi-Fi HaLow and two gateways for 802.15.4. For Wi-Fi, we see that the goals are attained for reliability, message latency and battery lifetime with 4 gateways. Therefore, an additional study for Wi-Fi with 5 gateways is considered. We observe that the number of alternatives to consider differs for each network technology. We notice that, unlike 802.15.4 and Wi-Fi, Wi-Fi HaLow manages to keep the reliability at 100% with one gateway, for both the minimal number of end-devices of 50 and the maximal number of end-devices of 100. We also see that, despite being the most performing in terms of battery lifetime and cost, Wi-Fi HaLow is outperformed by Wi-Fi and 802.15.4 in terms of message latency.

Table 5.7 shows that the **decision** step determines Wi-Fi HaLow with two gateways as being the best alternative among those considered. Figure 5.3 uses a radio chart to reflect the resulting KPIs after the application of the function f for the threshold-based KPIs (see Eqs. 5.3 and 5.4), the normalization and the weighting processes. Then, each resulting KPI value is divided by the maximum (for reliability and battery lifetime) or minimum (for message latency or cost) value of that KPI among all the alternatives. These values are finally plotted in the radio chart. Note that for readability purposes, we display the cost efficiency and latency efficiency, which are the inverse values of cost and message latency, respectively. Also, the "Latency Efficiency" axis is displayed using a logarithmic scale. We see that Wi-Fi HaLow, regardless of the number of gateways, significantly outperforms the other alternatives in terms of battery lifetime KPI. According to the calculated scores, the **decision** step returns Wi-Fi HaLow with 2 gateways as the network technology and topology to opt for the design phase of this application scenario.

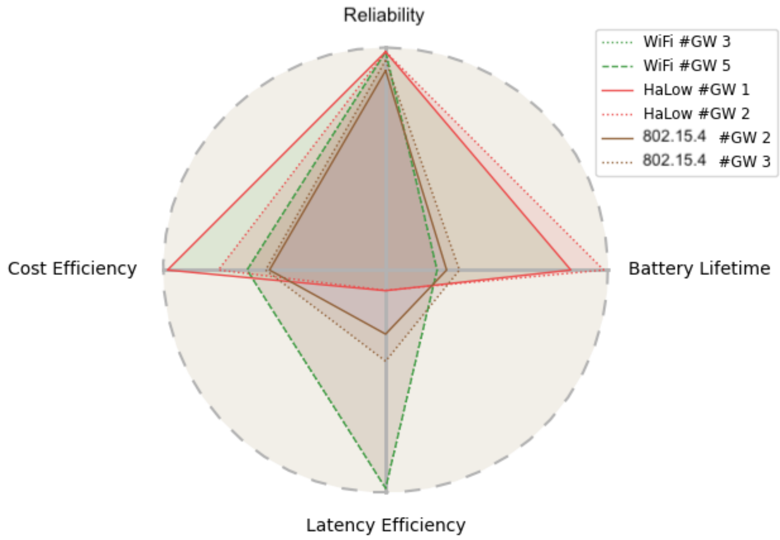


Figure 5.3: Case A: Smart Building Radio-chart.

5.4.2 Case B: Network Technology and Topology Selection for an Event Video-surveillance Application

We consider the design of a pre-packaged camera-based surveillance solution for events gathering large crowds (*e.g.*, trade fair, concert, etc.). Short installation time and stringent logistic constraints impose wireless connectivity with self-powered cameras placed at specific locations.

In the **application modeling** step, the application scenario is defined as follows: This solution is sold for a minimal (and maximal) number of 30 cameras, each one equipped with 2.4 Ah capacity batteries (powered by 3 V). The cameras typically send 200 packets of 1500 bytes per second to their gateway (leading to a workload of 2 Mbps). The application developers of the solution want to be able to improve the precision of the images. At maximal traffic workload, the cameras will be sending 300 packets every second to the gateway. This will result in a higher bandwidth requirement: A workload of 3 Mbps. The cameras are randomly placed around a gateway and the deployment scope is about 60 meters, in an outdoor urban environment. For the KPIs threshold, the architect specifies a reliability above 95%, batteries to last at least a week, and a message latency under 10 ms. The parameter l (expected lifetime) is set to two months, including the deployment and operations phases. The application models of Case Study B are available in Table 5.8.

Due to the large expected workload (2 and 3 Mbps), the **pre-selection** step dismisses LoRaWAN, Wi-Fi HaLow and 802.15.4 from the possible network technologies candidates. Thus, only Wi-Fi and 5G mmWave remain.

Regarding the **scenario design** step, the existing Wi-Fi configuration is the same as for case A, whereas 5G mmWave uses a 5G NR numerology of 2, disabled HARQ and RLC-AM.

At the **evaluation** step, the traffic workload defined in the **application modeling** step leads to a simulation time of around 2.5 seconds. HINTS iterates on the number of gateways. In this example, two gateways are

Application modeling	Parameters	Case B
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	30 30 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream 1500 200 300
Environment	• Type • Scope (meters) • Expected lifetime (days)	Urban 30 60

Table 5.8: Application Modeling of Case B.

enough for Wi-Fi to reach the KPIs goals of the threshold-based KPIs. Thus, we consider three alternatives for Wi-Fi with a number of gateways ranging from one to three. Regarding 5G mmWave, the improvement obtained when augmenting the number of gateways from 1 to 2 does not exceed ϵ, therefore three gateways are sufficient for the study (taking into account the safety margin). We present the results of the **evaluation** step in Table 5.9. First, we see that the reliability attains 100.0% starting from Wi-Fi with two gateways, while its value is around 55% for one gateway, which means that one gateway is not enough to support the whole traffic workload. It attains practically 100 % with one gateway for 5G mmWave. Also, we notice a slight increase in battery lifetime when the number of gateways is increased. Indeed, the more gateways, the less contention, resulting in end-devices spending more time in an idle state, which consumes less energy. The same remark cannot be made for 5G mmWave: The battery lifetime is not impacted by the number of gateways. Moreover, the obtained battery lifetime does not even last 1 day, while it manages to go up to 13 days for Wi-Fi with two gateways. The high energy consumption in 5G mmWave seems in line with the work [176] in which the authors showed that the 5G mmWave has substantial energy and computing power. In the same way, message latency slightly decreases for the same reason (less contention) for both network technologies. Regarding the cost, it tends to get lower with the number of gateways for Wi-Fi, which is due to the associated decreasing number of battery replacements, contrarily to 5G mmWave, where the cost seems to increase.

[176]: Mazaheri et al. (2019), 'A millimeter wave Network for Billions of Things'

Overall, Table 5.9 shows that the Wi-Fi alternative with 3 gateways obtains the best score and outperforms the others. The trade-offs between the different KPIs are captured in Figure 5.4 (computed as in Section 5.4.2) for the two best alternatives of each network technology. Figure 5.4 clearly shows that the Wi-Fi alternative with 3 gateways outperforms the other alternatives in terms of battery lifetime and cost efficiency.

Table 5.9: Case B: Event Video-surveillance Results.

Network technology		Minimal deployment (2 Mbps)				Maximal deployment (3 Mbps) *Scalability factor: 1*				
Technology	Nb. of GW	**Reliability** *Weight: 1 Unit: % Goal: >95*	**Battery Lifetime** *Weight: 1 Unit: d Goal: >7*	**Message Latency** *Weight: 1 Unit: ms Goal: <10*	**Cost** *Weight: 1 Unit: $*	**Reliability** *Weight: 1 Unit: % Goal: >95*	**Battery Lifetime** *Weight: 1 Unit: d Goal: >7*	**Message Latency** *Weight: 1 Unit: ms Goal: <10*	**Cost** *Weight: 1 Unit: $*	**Score**
Wi-Fi	1	55.85	4.81	1.53	1050	22.23	4.21	2.86	1200	0.43
Wi-Fi	2	100.0	13.57	0.2	1100	99.99	9.7	0.22	1400	0.82
Wi-Fi	**3**	**100.0**	**17.37**	**0.21**	**1050**	**100.0**	**15.61**	**0.25**	**1200**	**0.99**
5G mmWave	1	99.96	0.14	1.01	64100	99.97	0.14	1.05	64100	0.35
5G mmWave	2	100	0.1	0.95	90100	100	0.1	0.97	90100	0.32
5G mmWave	3	100	0.13	1.0	68400	100	0.13	1.0	68400	0.34

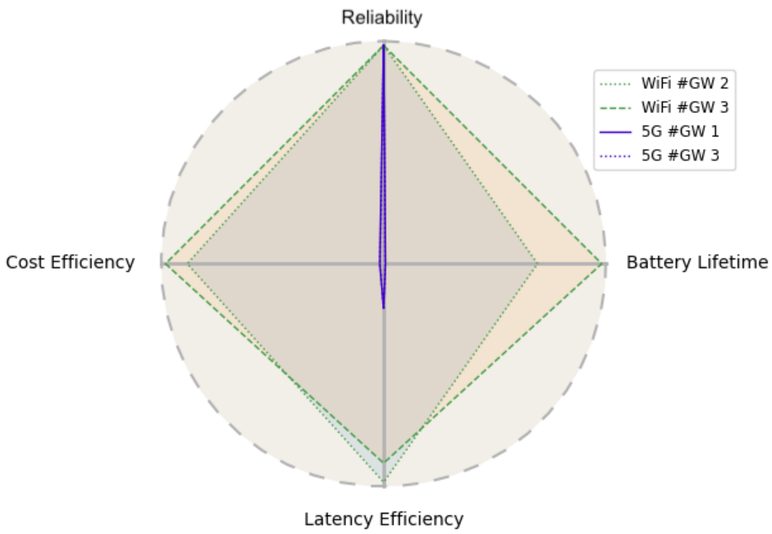


Figure 5.4: Case B: Event Video-surveillance Radio-chart.

5.4.3 Case C: Network Configuration Decision for a Precision Agriculture Application

In this case study, we consider the deployment of a pre-packaged precision agriculture solution in a given farm, using LoRaWAN. The solution architect needs to adjust the network configuration parameters of this solution taking into account the specificity of the deployment. The precision agriculture system comprises humidity, temperature and PH sensors, which measure these metrics before sending them to a LoRaWAN gateway for further transmission and processing.

At the **application modeling** step, the application scenario is defined as follows: The minimal number of sensors is 200, whereas the maximal number of sensors is 300, each one equipped with the same batteries as for Case A and B (2,400 mAh powered by 3 V). The sensors send one packet of 30 bytes every 3 minutes to their gateway, in an outdoor rural environment. The sensors are placed around the gateway and the deployment scope is 3,000 meters. Regarding the KPIs, this solution deployment typically requires a battery lifetime of 1 year, a message

Application modeling	Parameters	Case C
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	200 300 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream 30 0.005 0.005
Environment	• Type • Scope (meters) • Expected lifetime (days)	Rural 1500 3650

Table 5.10: Application Modeling of Case C.

latency lower than 1 second and a reliability of at most 90%. The value of l is set to 10 years. These parameters are displayed in Table 5.10.

Since the network technology to be used has already been decided (LoRaWAN), the **pre-selection** step is skipped.

In the **scenario design** step, the goal is to explore the various network configurations for the LoRaWAN settings within the end-devices. The goal is to determine which SF to select as well as which CR and type of traffic (confirmed or unconfirmed) to use[2]. Several network configurations are generated accordingly. We consider the minimal and the maximal values for each parameter.

2: A traffic type of 0 corresponds to an unconfirmed traffic, while 1 corresponds to a confirmed traffic.

In the **evaluation** step, the traffic workload leads to 36,000 seconds of simulation time to ensure a minimum of 200 packets sent by each sensor. This pre-packaged solution supports a single gateway. Table 5.11 presents the KPI values of the various LoRaWAN alternatives. First, we see that the SF has a tremendous impact on the KPIs, where the less the better: SF7 allows to ameliorate the performances almost by a factor of 2. Then, the type of traffic strongly influences the battery lifetime: In case the traffic is unconfirmed, it is around 8 times higher than with confirmed traffic. This is due to the re-transmissions triggered following up the loss of a packed when the traffic is confirmed.

In Table 5.11, the **decision** step elects LoRaWAN with a SF7, a 1 CR, with an unconfirmed traffic as the best alternative. The differences between the best alternatives' performances are shown in Figure 5.5 (computed as in Section 5.4.2).

Table 5.6 provides a summary of the **application modeling** for the case studies A, B and C. An outline of the rest of the steps is given in Table 5.12.

5.4.4 Discussion

The computational complexity of HINTS implementation mostly resides in the **evaluation** step, because running simulations (for example in

Table 5.11: Case C: Precision Agriculture Results.

Configuration			Minimal deployment (50 end-devices)				Maximal deployment (100 end-devices) *Scalability factor: 1*				
SF	Coding Rate	Traffic Type	**Reliability**	**Battery Lifetime**	**Message Latency**	**Cost**	**Reliability**	**Battery Lifetime**	**Message Latency**	**Cost**	**Score**
			Weight: 1 *Unit: %* *Goal: >90*	*Weight: 1* *Unit: d* *Goal: >730*	*Weight: 1* *Unit: ms* *Goal: <1000*	*Weight: 1* *Unit: $*	*Weight: 1* *Unit: %* *Goal: >90*	*Weight: 1* *Unit: d* *Goal: >730*	*Weight: 1* *Unit: ms* *Goal: <1000*	*Weight: 1* *Unit: $*	
7	**1**	**0**	**95.78**	**2560.7**	**82.17**	**12000**	**95.73**	**2560.7**	**82.17**	**17500**	**0.98**
7	1	1	99.44	331.37	82.17	112000	99.19	332.6	82.176	167500	0.51
7	4	0	93.65	1819.54	82.17	22000	93.74	1819.54	82.176	17500	0.81
7	4	1	98.65	232.95	82.17	152000	97.83	234.18	82.176	227500	0.5
12	1	1	43.47	126.44	197.4	282000	31.97	114.1	197.4	467500	0.0
12	1	0	43.31	126.44	197.4	282000	31.78	112.7	197.4	482500	0.38
12	4	1	33.29	126.77	197.4	282000	22.33	232.95	197.4	422500	0.38
12	4	1	33.11	126.77	197.4	282000	22.13	232.95	197.4	422500	0.38

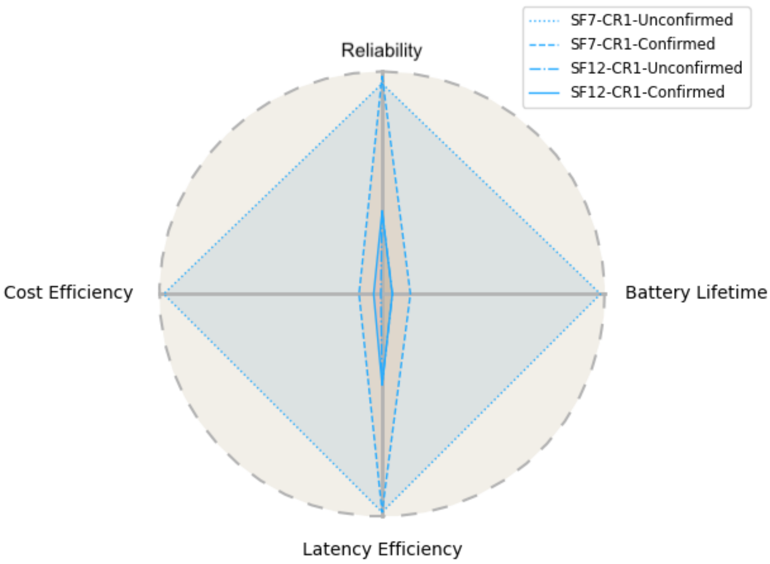


Figure 5.5: Case C: Precision Agriculture Radio-chart.

Table 5.12: Case Studies Summary.

Step / Case	Application modeling: Goals	Application modeling: Weights	Pre-selection	Scenario design	Evaluation	Decision
A	Reliability> 90% Lifetime> 80 days Latency< 100 ms	Reliability: 0.25 Lifetime: 0.25 Latency: 0.25 Cost: 0.25	Wi-Fi Wi-Fi HaLow 802.15.4	Single network configuration	See Table 5.7	Wi-Fi HaLoW with 1 GW
B	Reliability> 95% Lifetime> 7 days Latency< 10 ms	Reliability: 0.25 Lifetime: 0.25 Latency: 0.25 Cost: 0.25	Wi-Fi 5G mmWave	Single network configuration	See Table 5.9	Wi-Fi with 4 GW
C	Reliability> 90% Lifetime> 365 days Latency< 1000 ms	Reliability: 0.25 Lifetime: 0.25 Latency: 0.25 Cost: 0.25	LoRaWAN	$SF \in \{7, 12\}$ $CR \in \{1, 4\}$ Traffic type $\in$ {Unconfirmed (0), Confirmed (1)}	See Table 5.11	SF=7 CR=1 Traffic type= Unconfirmed

Table 5.13: Comparison of HINTS with the Related Work.

Reference	Considered KPI				Method to evaluate KPIs	Application driven	Automatic Selection
	Reliability	Message latency	Energy consumption	Cost			
Kanuch et al. (2020) [96]	-	-	✓	-	N/A	-	-
Sinha et al. (2017) [77]	-	✓	✓	✓	N/A	-	-
Ikpehai et al. (2018) [97]	✓	✓	✓	-	N/A	✓	-
Mekki et al. (2018) [59]	✓	✓	✓	✓	N/A	-	-
Mekki et al. (2019) [98]	✓	✓	✓	✓	N/A	-	-
Vejlgaard et al. (2017) [133]	✓	-	-	-	Simulation	✓	-
Lalle et al. (2019) [131]	✓	-	-	-	Simulation	✓	-
Verhoeven et al. (2022) [132]	✓	✓	-	-	Simulation	✓	-
Vannieuwenborg et al. (2018) [12]	-	-	-	✓	N/A	-	✓
Bari and Leung (2007) [134]	✓	✓	-	✓	N/A	-	✓
Bari and Leung (2007) [135]	✓	✓	-	✓	N/A	-	✓
Senouci et al. (2016) [136]	✓	✓	-	✓	N/A	-	✓
HINTS	✓	✓	✓	✓	Simulation	✓	✓

Network Simulator 3) can be time-consuming depending on the density of the simulated network and the simulation time. The **decision** step includes some relatively lightweight computations relating to the normalization, the weighting and the computing of the Euclidean distances (which results in a complexity of $O(n \times q)$, with q being the number of evaluated alternatives and n the number of KPIs). Overall, the **decision** step time is considered negligible in comparison to the evaluation time.

As for the issue of ranking reversal that may emerge in MADM methods, it does not apply in our case. Indeed, this classical problem refers to a change in the ordering among the alternatives, after the addition or the removal of an alternative from the group previously defined. For example, in the case of dynamic selection of a wireless interface, this alteration can affect the routing of packets [177]. Since HINTS targets static selection and gives recommendations prior to the effective network deployment, it is not subject to the ranking reversal problem. However, if we extend HINTS to address the emerging problem of dynamic reconfiguration of end-devices with multiple IoT networks, then we will have to deal with this issue.

[177]: Foubert et al. (2021), 'RODENT: A Flexible TOPSIS based Routing Protocol for Multi-technology Devices in Wireless Sensor Networks'

Table 5.13 shows that HINTS stands out from the other methods described in Section 3.2.2 in Chapter 3 by being an application-aware method and by providing an automatic selection mechanism at the same time. As for the KPIs, HINTS deals with the same set as many other existing works. This set includes the most important KPIs in the IoT networking field. Regarding the evaluation method to obtain the KPIs, most existing works ([12, 59, 77, 96–98, 134–136]) do not detail their way of evaluating the different network technologies. A number of works ([18, 131–133]) refer to discrete-event simulation, as HINTS does.

To end this discussion, we compare HINTS with the work of [12] which is the closest to our solution. Both methodologies help in modeling the network technologies and the IoT application. [12] provides questionnaires to help IoT architects eliminate technology candidates based on deployment constraints and on the evaluation of a single KPI, namely financial cost, for decision-making. On the other hand, HINTS relies on technical specification to eliminate network technology candidates but

[12]: Vannieuwenborg et al. (2018), 'Choosing IoT-connectivity? A guiding Methodology based on Functional Characteristics and Economic Considerations'

then resorts to an automated evaluation of KPIs (including the financial cost and network performance metrics) resulting from discrete-event simulation and decision algorithms to select the "right" network technology, its configuration and topology. We believe these two solutions (HINTS and [12]) are complementary and could be nicely combined in future works.

5.5 Conclusion

In this chapter, we have presented HINTS, a methodology enabling IoT network technologies selection and configuration. HINTS relies on the modeling of IoT network technologies on one side and a five steps decision process on the other side. We have described the different steps of this process, which include: (i) **Application modeling**, to abstract the IoT application specificity, its requirements on a set of KPIs; (ii) **pre-selection**, which dismisses the inappropriate network technologies; (iii) **scenario design**, to configure the application with the appropriate network technologies candidates, (iv) **evaluation** where the best-suited topology is iteratively found and an instrument such as simulation is used to estimate the KPIs on the targeted application scenario for each alternative and (v) **decision**, which assigns scores to each alternative using a MADM method, derived from TOPSIS. We have presented three case studies inspired by real-life deployments to illustrate the application of HINTS. The results have shown that HINTS enables a fair and insightful comparison of IoT network technologies for a given application scenario. Moreover, it permits to explore and determine network configuration parameters and the number of gateways to deploy. This work highlights the importance of the application context, the environment, and of the scaling factor in the network selection process and expected performance. For the sake of reproducibility, we made the source code available at [168].

In addition to the restriction on the star topology, there are other limitations for HINTS. First, the complexity of application and network modeling is still tangible. Indeed, since the targeted audience of a decision-support tool such as HINTS is primarily IoT architects, there is a real need of endowing them with the capacity to use these tools in the easiest way possible. This problem is tackled in Chapter 7. Then, since HINTS leans on simulation, the results it provides during the **evaluation** step require validation, to make sure that the simulator is grounded. Finally, the configuration optimization process, as seen in case study C, is still strongly tied to the generated scenarios for a given network technology. This means that the solution can only propose configurations that have been generated in the **scenario design** step. To have a more comprehensive comparison, it would be required to consider all the different possible configurations, which will naturally lead to an upsurge in the simulation time. A compromise between the comprehensiveness of the study and the execution time must be fined.

We address the last two limitations in the next chapter. First, we propose a method relying on Machine Learning (regression methods) to accelerate the design process using simulation, and we show an application for optimizing the configuration of IoT network technologies for a given application. Then, we propose a method to calibrate the energy consumption models from real deployments.

5.6 Appendix

5.6.1 Notation Table

Table 5.14 details the used symbols used throughout the chapter.

Table 5.14: Table of Notations.

Symbol	Signification
R	Application requirements
K	KPIs
G	KPIs performance goals
W	KPIs weights
T	Network technologies candidates
C	Network configuration parameters
P	KPIs values of the alternatives
N	Normalized KPIs
W	KPIs weights
V	Weighted normalized KPIs
V^+	Ideal positive solutions
V^-	Ideal negative solutions
S^+	Positive distances
S^-	Negative distances
q	Number of evaluated alternatives
ϵ	Minimum improvement for the KPIs
n	Number of KPIs
β	Scalability factor

6 Addressing the Limits of Simulation

6.1 Introduction

In the last chapter, we have proposed HINTS, a methodology to enable IoT network technologies selection and configuration. Since HINTS relies on simulation, it can lead to the same inconvenience and even aggravate them. These burdens are: Cost (in terms of time), accuracy, complexity and accessibility. We explore ways of addressing the first two limitations in this chapter, while complexity and accuracy are tackled in Chapter 7.

When considering the cost aspect, utilizing HINTS can enable one to make informed configuration decisions for example. However, this benefit comes with the trade-off of increased time required for decision-making processes. Indeed, the number of possible configurations for a network technology can be very important. For instance, SF for LoRa, which has a great impact on the performance, can take integer values between 7 and 12. Add to that the other configuration parameters of LoRa and their possible values, and the set of combinations becomes very quickly unmanageable. Yet, determining the appropriate values to use is a key issue for IoT architects when deploying their solution, especially since several KPIs can be considered (energy, throughput, latency, etc.). Even though simulation offers relative scalability, testing all the configuration combinations could become costly in terms of time and energy consumption. There is a need to reduce the number of simulations to reduce these two factors.

Regarding accuracy, NS carries the need for validation and calibration to make sure that it returns accurate results [178]. For instance, energy efficiency is a critical aspect of IoT network technologies, as many devices operate on limited battery power or in energy-constrained environments. The simulation models can yield unreliable results. Accurate energy consumption models ensure that the simulated results align closely with the actual energy usage of the deployed IoT devices and networks. Thus, this will make the simulation more trustworthy for IoT architects.

[178]: Pawlikowski et al. (2002), 'On Credibility of Simulation Studies of Telecommunication Networks'

In this chapter, we propose two methods to address the limitations mentioned above. First, to reduce simulation time, we propose COSIMIA (COmbining SIMulatIon and mAchine learning), which relies on machine learning to exploit only a part of the exploration space for the design decision process. As an example of application, we show how combining simulation and Machine Learning (ML) can help to optimize the configuration of a network technology for a given IoT scenario, with a reduced number of simulations. Then, we propose a calibration process that can be applied to improve the accuracy of simulation results, regarding energy consumption. We demonstrate the feasibility of calibrating simulation energy models using linear regression based on real measurements with a Proof of Concept (PoC), highlighting its effectiveness in improving the accuracy of the simulation models.

The remainder of this chapter is divided in two parts: Part 6.2 is about the reduction of simulation time for the design decision, with an application on configuration optimization. In this part, Subsection 6.2.1 provides a formulation of the tackled problem, while the proposed method is described in Subsection 6.2.2. Examples of application are presented in Subsection 6.2.3.

Part 6.3 concerns the simulation accuracy. The problem is formulated in Subsection 6.3.1. We describe our solution in Subsection 6.3.2 and we show an example of application using ns-3 and the FIT IoT-Lab testbed in Subsection 6.3.3. Related works on energy calibration in simulation are presented in Subsection 6.3.4.

The conclusion of the whole chapter is provided in Section 6.4.

6.2 Simulation Time

6.2.1 Problem Formulation

We formulate the design decision acceleration process problem like the following. We have the following parameters as inputs:

- An IoT application A, with its set R of characteristics and communication requirements.
- A network technology T.
- A set C of possible parameter combinations.
- A set K of key performance metrics or KPIs that characterize the behavior of an application A on a network technology T with C.

The decision problem consists in finding the parameters C_d in C of the network technology T that fit the application requirements R and provide the best performances for the application A, in terms of KPIs K.

6.2.2 Proposed Solution

As we have seen in the previous chapter, HINTS can help solve this problem. However, in its initial version, it would need a high number of simulations. Indeed, the number of possible parameter combinations for network settings (including generic and specific parameters) can be important. If we consider that there are n different parameters, where each one can take m different values, a comprehensive research would lead to m^n simulations. Added to that the time that each simulation takes, this can quickly become overwhelming.

Thus, the idea of COSIMIA is to generate a reduced sample of the exploration space (the whole set of possible combinations), calculate a score for each one and use regression models to learn the scoring of each combination. Finally, instead of using real simulation, we make a comprehensive inference using the trained regression models to find the best combination. These steps are depicted in Figure 6.1, and we detail them in what follows:

Data generation: This step consists in generating a sample of parameter combinations, and run simulation to calculate the resulting KPIs for each

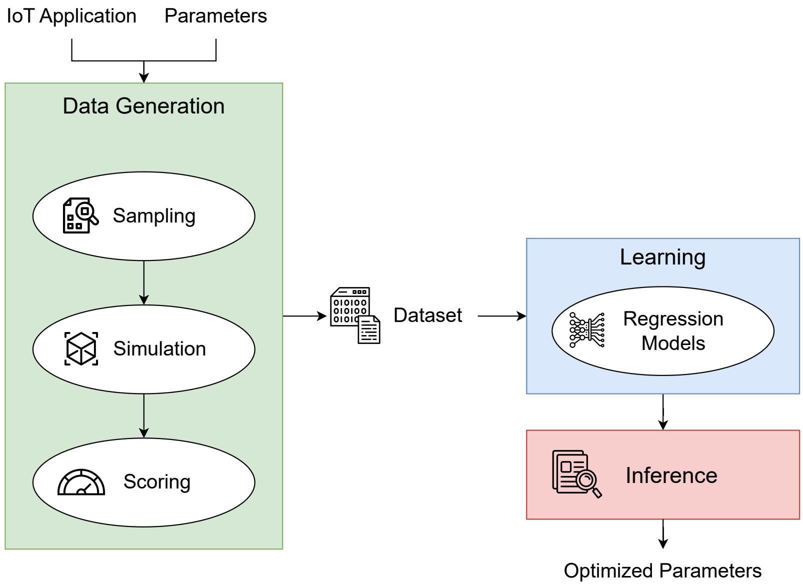


Figure 6.1: COSIMIA functioning.

combination. The sampling is done as the following: For each parameter, we consider the minimal, middle and maximal values (*e.g.*, the values 7, 10 and 12 for SF).

Then, these different parameter combinations are tested using simulation, using ns-3 in our case. The resulting KPIs are then gathered from the simulation. Finally, a score is assigned to each parameter combination, representing the relevance of the parameter combination, using MADM methods [179] (as done in HINTS, Section 5.3.6 of Chapter 5). This way, and compared to a comprehensive research, still considering n parameters with m possible values for each one, this step requires at most 3^n simulations.

At the end of this step, we obtain a dataset composed by different parameter combinations, where each one has a score based on the its resulting KPIs.

Learning: The second step consists in applying regression models[1] to predict the score of a given parameter combination, without using simulation. This is done by feeding the generated dataset of samples to different regression models, where the input variables are the network parameter combinations, and where the output is the score. Thus, we train models to learn to predict the correct score of a given parameter combination. Figure 6.2 shows an example of a generated dataset, highlighting the inputs and the output of the regression models.

1: Regression is a technique for investigating the relationship between independent features and a dependent outcome. It is a branch of machine learning, where algorithms are used to predict continuous outcomes from given input features.

Inference: Once the models have been trained, a comprehensive inference is conducted for all the possible parameter combinations. For each one, instead of running the simulation, we use the trained models to calculate the score. Note that the comprehensive inference is made possible because the prediction of the score is fast, compared to the actual simulation.

Decison: The decision step consists in comparing the different parameter combinations according to their score. The parameter combination which returns the best score is the one retained.

Input   Output

nGW	sf	traffic_type_	coding_rate	crc	success_rate	energy	latency	price	score
3	8	1	2	1	91,25	0,00223759	238,08	3000	0,69705853
3	9	0	2	1	85,2	0,00050719	427,008	3000	0,67935014
3	9	1	2	1	97,47	0,00231686	427,008	3000	0,68267309
3	10	0	2	1	89,38	0,00089007	755,712	3000	0,79743684
3	10	1	2	1	100	0,00239554	755,712	3000	0,65457617
3	11	0	2	1	94,38	0,00199821	1708,03	3000	0,58826983
3	11	1	2	1	93,55	0,00219818	1708,03	3000	0,57321259
3	12	0	2	1	88,39	0,00247255	3022,85	3000	0,41632534
3	12	1	2	1	88,39	0,00247255	3022,85	3000	0,41632534
4	7	0	2	0	83,58	0,00015571	125,184	4000	0,69048149
4	7	1	2	0	95,83	0,00121376	125,184	4000	0,79111444
4	8	0	2	0	84,17	0,00027286	225,792	4000	0,8493187
4	8	1	2	0	100	0,00209729	225,792	4000	0,70091466
4	9	0	2	0	91,67	0,0004786	402,432	4000	0,83734485
4	9	1	2	0	100	0,00190094	402,432	4000	0,71031007

Figure 6.2: Regression Models Input and Output.

Application on Configuration Optimization

In the following, we will focus on configuration parameters of network technologies (*e.g.*, SF for LoRa or Modulation and Coding Scheme (MCS) for Wi-Fi). The objective is to be able to provide a decision for the configuration of an IoT network technology by reducing the number of simulations, *i.e.*, without making a comprehensive simulation for all the different configuration parameter combinations. We detail in what follows the used concepts by COSIMIA for the configuration optimization:

Application modeling: We make the choice in our study to model the application by the following parameters (based of the application modeling provided in 4.3.1):

(i) Number of end-devices.
(ii) Message size, in bytes.
(iii) Message period, which is the interval of time between each message transmission, in seconds.
(iv) Deployment scope, which represents the maximal distance between two end-devices.
(v) Radio environment, which expresses the radio conditions where the IoT application is deployed. It can be rural, urban, suburban or indoor.

We consider that these five parameters are sufficient in our study to have a primary characterization of an IoT application.

Network technology configuration: As mentioned in Section 6.1, each IoT technology is characterized by a set of parameters that are often unique to each IoT technology. In addition, we consider the number of gateways as a common parameter for all technologies (with a star topology) that needs to be optimized. We define a network configuration by a combination of these parameters (including the number of gateways). Note that regarding the number of gateways, the sampling is done by iterating it from 1 to a maximal value that we set to the number of end-devices divided by 5[2], with a step of 3.

2: This value is arbitrary and could be differently set.

KPIs: From the KPIs presented in Section 4.3.4, we consider:

- **Reliability**, which represents the amount of correctly received packets among all the sent ones.
- **Energy consumption**, which is the amount of energy consumed by the end-devices during the network deployment.

- **Packet latency**, which is the average time that packets take to flow from the end-device to the gateway.
- **Cost**, which is an estimation of the cost of deployment of the network. It represents in our case the purchase cost of the gateways.

Note that any other KPI, such as security or environmental impact, can be considered. If it is a network-related KPI (*e.g.*, jitter), the only condition is that the used network simulator allows to compute it.

Score computing: A score is used to evaluate each network configuration, based on its obtained KPIs through the simulation. Finding the best network configuration amounts to finding the one with the highest score. As for HINTS, we use Technique for Order Preference Similarity to Ideal Solution (TOPSIS) to compute the score. Only, in this case, the alternatives are the network configurations, while the attributes are the considered KPIs.

Regression models: Several models have been proposed in the literature for tackling the regression problem. We use the following regression models, which are extensively used by the community :

- **Linear Regression**: It is an approach for predicting a quantitative response Y on the basis of input variables. It assumes that there is approximately a linear relationship between the input variable and the output Y [180].
- **Gradient Boosting**: It builds a set of prediction models in a sequential manner. It combines these models to create a more efficient predictive model than the initial ones. The technique works by fitting the models to the residuals or errors of the previous models, with each subsequent model focusing on reducing the errors made by the previous models. The final prediction is made by aggregating the predictions of all the models [181].
- **Random Forests**: It works by constructing a set of decision trees using random subsets of the training data and random subsets of the features. Each decision tree in the set independently predicts the target variable, and the final prediction is made by aggregating the predictions of all the trees [182].
- **Extra Trees**: It builds multiple decision trees using random subsets of the training data and features. However, unlike random forests, Extra Trees further randomizes the splitting process by considering random thresholds for each feature instead of searching for the best split point [183].
- **K-nearest Neighbors**: In k-NN, the prediction for a new data point is based on the majority vote or averaging of the k nearest neighbors in the training data. The distance metric, such as Euclidean distance, is used to measure the similarity between data points [184].
- **Support Vector Regression**: SVR aims to find a hyperplane in a high-dimensional feature space that has the maximum margin from the training data points. The algorithm tries to minimize the error between the predicted and actual values while allowing a certain degree of tolerance [185].

[180]: Seber et al. (2003), *Linear Regression Analysis*

[181]: Friedman (2001), 'Greedy Function Approximation: A Gradient Boosting Machine'

[182]: Breiman (2001), 'Random Forests'

[183]: Geurts et al. (2006), 'Extremely Randomized Trees'

[184]: Peterson (2009), 'K-Nearest Neighbor'

[185]: Smola et al. (2004), 'A Tutorial on Support Vector Regression'

An algorithm of COSIMIA for the configuration optimization is given in Algorithm 6.

Algorithm 6: COSIMIA Algorithm

Inputs:
$R = [R_1, \ldots, R_k]$; Application requirements;
T: IoT network technology;
$C = [c_1, \ldots, c_n]$; $c_i \in [a_i, \ldots, b_i]$; $c_i, a_i, b_i \in \mathbb{R}$; Configuration parameters;
$N_{ED} \in \mathbb{N}$; Number of end-devices;
$N_{GW} \in \mathbb{N}$; Number of gateways;
N_S: Network Simulator;
$K = [K_1, \ldots, K_m]$, $K_i \in \mathbb{R}$; KPI values;
S: Scoring function; $S : \mathbb{R}^n \longrightarrow \mathbb{R}$;
M: Regression model;
D: Configuration samples dataset;
Algorithm:
/* Initialization */
$D \longleftarrow \varnothing$
/* Sampling */
for N_{GW} in $[1, \frac{N_{ED}}{5}, 2]$ **do**
    for c_1 in $\{a_1, \frac{b_1+a_1}{2}, b_1\}$ **do**
        ...
        for c_n in $\{a_n, \frac{b_n+a_n}{2}, b_n\}$ **do**
            $K \longleftarrow N_s(R, T, C)$
            $D.insert(C, K)$
        end for
        ...
    end for
end for

/* Scoring */
for (C, K) in D **do**
    $(C, K) \longleftarrow (C, K, S(K, D))$
end for

/* Learning */
$M.learn(C, S(K))$

/* Inference */
$best_{config} \longleftarrow [c_1, \ldots, c_n]$
$best_{score} \longleftarrow 0$
for N_{GW} in $[1, \frac{N_{ED}}{5}]$ **do**
    for c_1 in $[a_1, b_1]$ **do**
        ...
        for c_n in $[a_n, b_n]$ **do**
            if $M.predict(C) > best_{score}$ **then**
                $best_{config} \longleftarrow C$
                $best_{score} \longleftarrow M.predict(C)$
            end if
        end for
        ...
    end for
end for

return $best_{config}$

Application modeling	Parameters	Case A
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	200 200 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream 50 0.001 0.001
Environment	• Type • Scope (meters) • Expected lifetime (days)	Rural 8000 N/A

Table 6.1: Application Modeling of Case A.

6.2.3 Examples of Application

In this section, we show the application of COSIMIA for the configuration optimization with three network technologies: LoRa, Wi-Fi (802.11ac) and 802.15.4 (6LoWPAN).

Case Study A: Smart Agriculture using LoRa

Based on Section 4.3.2, the considered parameters for LoRa are:

- **SF:** Determines the speed at which the signal frequency changes across the bandwidth of a channel. The higher the spreading factor the lower the data rate.
- **CR:** An indication of how much of the data stream is actually being used to transmit usable data.
- **CRC:** An error-detecting code commonly used in networks to detect accidental changes in the transmitted data.
- **Type of traffic:** Determines whether the data is sent with or without an acknowledgement. It can therefore be confirmed (1) or unconfirmed (0), respectively.

The implemented IoT application can be assimilated into a smart agriculture solution, defined as follows: 200 sensors send 50 bytes packets every 600 seconds (10 minutes) to the gateways. The sensors are separated by a distance (deployment scope) of 8000 meters and are deployed in a rural environment. These parameters are summarized in Table 6.1.

Table 6.2 shows the results obtained for this first case study. The comprehensive simulation shows that the optimal solution is to use 5 gateways, a SF of 8, unconfirmed traffic, a CR of 1 and a CRC of 0. Determining this solution required no less than 441 minutes and 3840 simulations, while COSIMIA required 70 minutes and 480 simulations. This is equivalent to a reduction in simulation time by a factor of 6. As for the proximity of the regression models, it is defined by the ratio of the score of the solution returned by each model to the optimal solution (returned by the comprehensive simulation)[3]. We find that most of the models reach 99%

3: Note that for the considered case studies, although it takes considerable time, it is still possible to execute the simulations for all the different combinations. The comprehensive simulation is conducted to validate the regression model outcome

Table 6.2: Results for Case Study A. The format of the solutions is the following: [NGW,SF,Traffic-Type,CR,CRC].

Model	Solution	KPIs				Data generation		Proximity
		Reliability (%)	Energy consumption (Mili-Watts)	Latency (ms)	Cost ($)	Time (minutes)	Number of simulations	
Comprehensive simulation	[5,8,0,1,0]	99	0.082	195	5000	441	3840	N/A
Gradient boosting	[5,7,0,1,1]	89.5	0.093	112	500	70	480	0.99
Extra trees	[5,7,0,1,0]	89.5	0.048	107	5000			0.99
Random forest	[5,7,0,1,1]	89.5	0.05	112	5000			0.99
KNN	[5,8,0,2,1]	99	0.082	195	5000			0.997
SVR	[10,7,0,1,1]	100	0.048	107	10000			0.94
Linear regression	[1,7,0,1,0]	4.45	0.047	107	1000			0.81

of proximity. However, the linear regression struggles to exceed 85% of proximity, which corroborates the fact that the problem is not linear.

Case Study B: Video-surveillance using Wi-Fi

Still based on Section 4.3.2, the considered parameters for Wi-Fi are:

- **Number of spatial streams (SI):** Determines the number of streams where coded data signals can be sent and received independently.
- **Packet aggregation (PA):** Determines whether packet aggregation, which is the process of joining multiple packets together into a single transmission unit, is enabled or disabled.
- **Short Guard Interval (SGI):** Is the space between symbols (characters) being transmitted. It can either be short (0.4 µS) or long (0.8 µS).
- **MCS:** An index based on several parameters of a Wi-Fi connection between two stations. Namely, for 802.11ac, it depends on the modulation type, the CR, the number of spatial streams, the channel width, and the guard interval.

We consider in this case an event video-surveillance use-case, where 30 cameras send 2000 bytes packets every 5 milliseconds (which leads to a data rate of 3 Mbps) to the gateways. The cameras are deployed in the vicinity of the gateways, with deployment scope of 30 meters, in a suburban environment. Table 6.3 summarizes these parameters.

We see in Table 6.4 that the simulation time is reduced from 450 to 55 minutes, with ten times fewer needed simulations (720 vs. 72). Most regression models are able to propose configurations that are at 95 % proximate from the optimal solution returned by the comprehensive simulation. Once again, the linear regression model has difficulty returning an interesting configuration in terms of proximity from the optimal. Indeed, the returned solution is only at 80% proximate from the optimal.

Case Study C: Telemetry using 802.15.4

As presented in Section 4.3.2, the considered parameters for 802.15.4 (6LoWPAN) are:

Table 6.3: Application Modeling of Case B.

Application modeling	Parameters	Case B
End-devices	• Minimal number	30
	• Maximal number	30
	• Battery capacity (Amperes.hour)	2.4
Workload	• Traffic direction	Upstream
	• Message size (bytes)	2000
	• Minimal frequency (packets/second)	200
	• Maximal frequency (packets/second)	200
Environment	• Type	Suburban
	• Scope (meters)	30
	• Expected lifetime (days)	N/A

Table 6.4: Results for Case Study B. The format of the solutions is the following: [NGW,MCS,PA,SI,SGI].

Model	Solution	KPIs				Data generation		Proximity
		Reliability (%)	Energy consumption (Watts)	Latency (ms)	Cost ($)	Time (minutes)	Number of simulations	
Comprehensive simulation	[2,0,0,2,1]	100	0.03	0.09	200	450	720	N/A
Gradient boosting	[3,1,0,3,0]	100	0.02	0.06	300	55	72	0.95
Extra trees	[3,1,0,3,0]	100	0.02	0.06	300			0.95
Random forest	[3,1,0,3,0]	100	0.02	0.06	300			0.95
KNN	[3,7,0,3,0]	100	0.02	0.06	300			0.95
SVR	[5,1,1,2,0]	100	0.02	0.12	500			0.83
Linear regression	[6,9,1,1,1]	100	0.02	0.12	600			0.8

- **Number of frame retries (FR):** It is the number of the retransmissions limit when there is no acknowledgement received before dropping the packet.
- **CSMA backoff (BE):** The number of times that the node stays in the backoff stage after unsuccessful channel sensing.
- **Maximal backoff exponent (MinBE):** Maximal random interval before sensing the channel.
- **Minimal backoff exponent (MaxBE):** Minimal random interval before sensing the channel.

In this case, we consider a telemetry use-case, where 50 sensors send 100 bytes packets every second to the gateways. The sensors are separated by a distance of 200 meters in a suburban environment. Table 6.5 summarizes these parameters.

Table 6.6 shows that the improvement in execution time is much larger for this case study. Indeed, we go from 1367 minutes for the comprehensive simulation to 26 minutes for the data generation, which is equivalent to an improvement of a factor of 60. This is due to the fact that the configuration parameters specific to 802.15.4 have on average higher cardinalities than the configuration parameters specific to LoRa. Still, the proximity is also on the order of 99% for most of the models, while the linear regression model is still underperforming.

Table 6.5: Application Modeling of Case C.

Application modeling	Parameters	Case C
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	50 50 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream 100 1 1
Environment	• Type • Scope (meters) • Expected lifetime (days)	Suburban 200 N/A

Table 6.6: Results for Case Study C. The format of the solutions is the following: [NGW,MaxBE,MinBE,CB,FR].

Model	Solution	KPIs				Data generation		Proximity
		Reliability (%)	Energy consumption (Watts)	Latency (ms)	Cost ($)	Time (minutes)	Number of simulations	
Comprehensive simulation	[3,4,3,4,0]	92	0.03	5.56	300	1367	23040	N/A
Gradient boosting	[3,5,3,5,3]	99.37	0.032	5.58	300	26	405	0.99
Extra trees	[3,6,0,0,4]	93.75	0.031	3.91	300			0.99
Random forest	[3,8,7,5,3]	100	0.033	31.16	300			0.98
KNN	[3,8,7,5,6]	100	0.033	31.16	300			0.98
SVR	[5,7,7,5,6]	100	0.03	30.15	500			0.94
Linear regression	[10,8,7,5,7]	100	0.02	26.02	1000			0.79

As we have seen for use-cases A, B and C with three different applications and featuring three different network technologies, COSIMIA has been able to return optimized configurations which are close from the optimal one returned by a comprehensive simulation. Moreover, this has been possible through a clear reduction of the simulation time, with factors of 6, 10 and 60 for use-cases A, B and C, respectively.

Impact of the sampling granularity: In the following, we investigate the impact of the sampling granularity on the performance of the method. The average proximity is the mean of the proximities of all the tested regression models.

Figure 6.3[4] shows that the finer the granularity (in other words, the more points are taken for sampling) the longer the execution time. This is of course due to the fact that the generation of the data requires more simulations, which are themselves time-consuming. Regarding the proximity of the returned solution, it is around 80% when choosing only two points for the sampling (minimum and maximum values for each parameter). From granularity 3 (minimum, intermediate and maximum values as we recommend in COSIMIA) the average precision reaches 95% and remains fixed at this value for the higher values.

4: Note that "Comp-Sim" corresponds to the comprehensive simulation done to all the possible configurations.

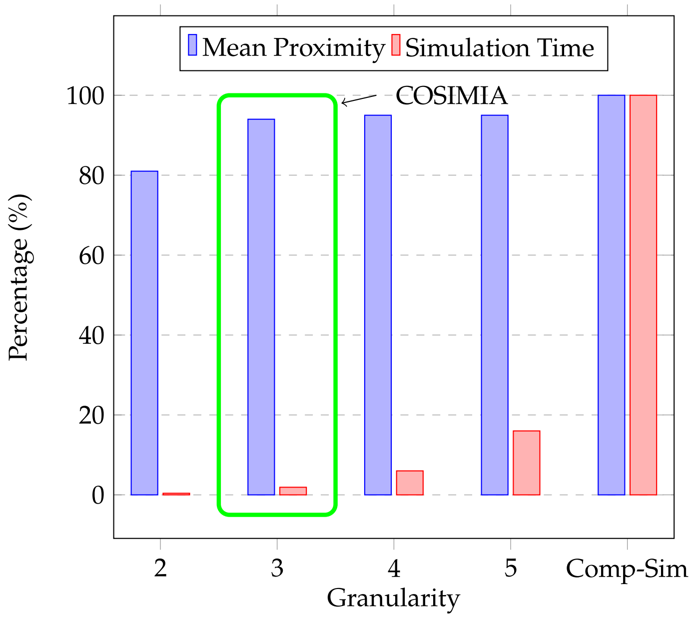


Figure 6.3: Impact of the Sampling Granularity of the Solution.

6.3 Simulation Accuracy

6.3.1 Problem Formulation

Calibration can be seen as a refining of the simulation models in order to produce more realistic simulations. The only KPI on which we focus here is the energy consumption. The energy consumption is highly tied not only to the used network technology but also to the very deployment. Indeed, the radio conditions, the nodes positions and the application workload can strongly influence the energy consumption, as we have seen through the models presented in Section 4 in Chapter 3. For this reason, the energy calibration can be done in case we have a small scale Proof of Concept (PoC) with real devices. As a recall (see Section 3.2.1 in Chapter 3), energy consumption in discrete-event network simulators is modeled as follows:

$$E = \sum_{i \in S} (\alpha_i \times t_i) \times V \tag{6.1}$$

where:

- E: Energy consumption in Joules,
- S: Set of different physical states,
- α_i: Current consumption of state i in Amperes,
- t_i: Total time passed a state i in Seconds,
- V: Voltage in Volts.

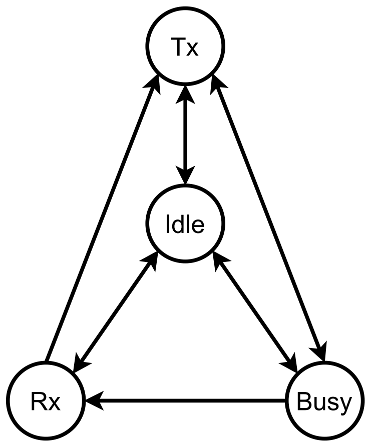


Figure 6.4: Wi-Fi NIC Physical State Machine in ns-3.

As examples, Figure 6.4 depicts the physical states of Wi-Fi NIC in ns-3, and the possible transitions between them (Further details about this model can be found in [156]), while Table 6.7 indicates the numerical values we used throughout the thesis to compute the energy consumption of the Wi-Fi, LoRaWAN, Wi-Fi HaLow and 802.15.4 NIC. The values for Wi-Fi were selected by calibrating the state machine against the measurements provided by Serrano et al. in [130]. The values for LoRaWAN are

[156]: Wu et al. (2012), 'An Energy Framework for the Network Simulator 3 (ns-3)'

[130]: Serrano et al. (2014), 'Per-frame Energy Consumption in 802.11 Devices and its Implication on Modeling and Design'

those given by default in the ns-3 module for the LoRaWAN consumption by Magrin et al. [155]. For 5G mmWave, the values are derived from [186]. For all these different network technologies, we did not take into account the type of the used end-devices (camera, sensor, etc.). For this reason, the obtained results for energy consumption may be relatively unreliable.

[155]: Magrin et al. (2017), 'Performance Evaluation of LoRa Networks in a Smart City Scenario'

[186]: Sen et al. (2021), 'An ns3-based Energy Module of 5G NR User Equipments for Millimeter Wave Networks'

Table 6.7: Drawn current values (in mA) for each state of the machine state used in ns-3 simulations to evaluate the power consumption of LoRa, 5G mmWave, Wi-Fi HaLow, Wi-Fi and 802.15.4 communications.

State \ Technology	LoRaWAN	5G mmWave	Wi-Fi HaLow	802.15.4	Wi-Fi
Tx	77	350	7.2	7	107
Rx	28	350	4.4	1.5	40
Idle	1	/	1	/	1
Sleep	0.015	45	/	/	/
CCA Busy	/	1	/	/	/
Switch	/	/	/	0.5	/

The major problem with the modeling of Equation 6.1 is that, in reality, the current consumption α_i of each state i is strongly tied to the type of equipment. Indeed, although several works associate network technologies to current consumption values (*e.g.*, [77], [64]), it is possible to find different equipment featuring the same network technology with different current consumption values (*e.g.*, in [187], [128]).

[77]: Sinha et al. (2017), 'A Survey on LPWA Technology: LoRa and NB-IoT'

[64]: Noreen et al. (2017), 'A Study of LoRa Low Power and Wide Area Network Technology'

[187]: Gomez et al. (2019), 'A Sigfox Energy Consumption Model'

[128]: Casals et al. (2017), 'Modeling the Energy Performance of LoRaWAN'

6.3.2 Proposed Solution

Since we consider that we have a small scale PoC of the deployment to make our calibration, we suppose that we have access to the sensors energy consumption. Thus, we propose to used real measures to calibrate the α_i values, as follows, and as depicted in Figure 6.5:

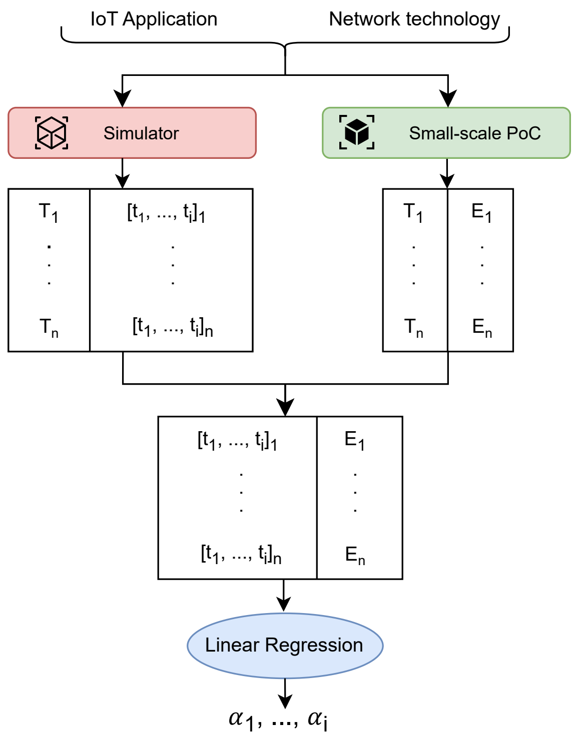


Figure 6.5: Calibration Method functioning.

1. Take periodic measurements of the power consumption at a regular pace.
2. For fixed periods, calculate the energy consumed. To do so, one can use the integral of the power per time (seconds)[5]. Generate a dataset D_1.
3. For the same considered period, generate a trace of all the crossed states in the simulator (Tx, Rx, etc.) and the corresponding times passed in each state. Generate a dataset D_2.
4. Merge D_1 and D_2, so that we have for each period the amount of time passed in each state in the simulator, and the consumed energy in the physical network.
5. Apply a linear regression to infer the coefficients by which the times passed in the different states must be multiplied to get the real energy consumed (after multiplying with the voltage that we consider fixed). These coefficients correspond to the calibrated α_i[6].

5: The integral can be calculated using Thomas Simpson's method.

6: Since the coefficients must be positive, we use the least square method for the linear regression.

The pseudo-algorithm of the solution is presented in Algorithm 7.

6.3.3 Example of Application

Use-case Description

Let us consider the case of IoT devices placed in manufacturing facilities that transmit sensor data, in order to monitor critical variables like temperature, pressure, or machine vibrations. The objective is to enable real-time monitoring of production processes, early detection of equipment failures or abnormalities, and proactive maintenance to prevent downtime and optimize operational efficiency. We suppose that the end-devices are connected to a measuring platform capable of measuring the energy consumption of each node.

Before deploying this solution, IoT architects would like to have a small-scale PoC of the deployment to be able to conduct what-if analysis and scenario simulations during the design phase. This would allow them to assess the performance of different configurations to select the most relevant one. This helps in evaluating the effectiveness of potential improvements before implementing them in a real-world environment. Besides that, the IoT architects are concerned about the relevance of the simulation models of the small scale PoC. They would like to be sure that the provided results are grounded and close to reality. For this reason, during the ongoing management phase of their IoT solution, they would also like to use real measures of energy in order to calibrate the simulation models of the PoC.

We consider that there are 50 sensors sending 100 bytes packets every second to a gateway. The sensors are separated by a distance of 200 meters. These parameters are available in Table 6.8. The communication are made through IEEE 802.15.4 network technology.

To illustrate the application of our method, we use the FIT IoT-Lab and ns-3 as the real deployment platform and the network simulator, respectively. For the end-devices, we use the M3-board micro-controllers (see Figure 6.6), and we implement the firmware using RIOT [53] (See Section 2.3 in Chapter 2).

[53]: Baccelli et al. (2013), 'RIOT OS: Towards an OS for the Internet of Things'

Algorithm 7: Energy Consumption Calibration Algorithm

Inputs:
$R = [R_1, \ldots, R_k]$; Application requirements;
T: IoT network technology;
$Time$: Deployment time;
$P(t)$: Mesured power consumption at instant t;
$period$: Power consumption measurement period;
δ: Energy consumption calculation period;
N_S: Network Simulator;
P: Power consumption dataset;
D_1: Energy consumption dataset;
D_2: Physical states times dataset;
D: Final dataset;
Algorithm:
/* Initialization */
$D \leftarrow \varnothing$
/* Measurement */
while $t \leq Time$ **do**
  $P.insert(t, P(t))$
  $t \leftarrow t + period$
end while

/* Energy Calculation */
$t \leftarrow 0$
while $t \leq Time$ **do**
  $E \leftarrow \int_t^{t+\delta} M(t)\, dt$ /* Consumed energy between t and $t + \delta$ */
  $D_1.insert(E)$
  $t \leftarrow t + \delta$
end while

/* Physical State Times in Simulation */
$logs \leftarrow S(R, T, N_S).logs$ /* Logs of the simulated deployment (containing physical states and corresponding times) */
$t \leftarrow 0$
while $t \leq Time$ **do**
  $times \leftarrow logs[t, t + \delta]$ /* Physical state times between t and $t + \delta$ */
  $D_2.insert(times)$
  $t \leftarrow t + \delta$
end while

/* Regression */
$D \leftarrow merge(D_1, D_2)$
$LinearReg(times, E)$

return $LinearReg.coefficients$

Application modeling	Parameters	Case Study
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	40 60 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream 100 1 2
Environment	• Type • Scope (meters) • Expected lifetime (days)	Suburban N/A N/A

Table 6.8: Application Modeling of the Case Study.

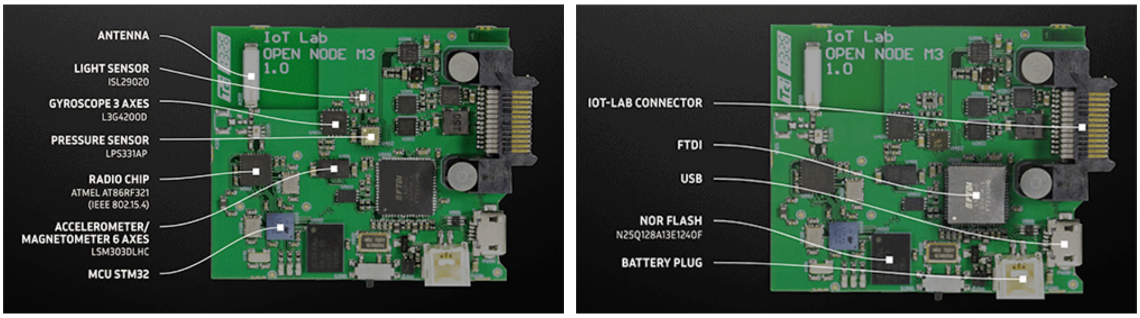


Figure 6.6: M3-board Micro-Controller Components [99].

Results

Table 6.9 shows the drawn current values of the simulation models before and after the calibration. As we can see, the values are clearly different for most of the NIC physical states.

To highlight this difference, Figure 6.8 displays the power consumption (in milliwatts) measured for one device on the real deployment (FIT IoT-Lab), through the calibrated simulation models and the default simulation models of ns-3. As we can see, the calibrated models manage to reproduce in a very accurate way the power consumption measured on the real deployment. It is also interesting to see that the power consumption measured through default models is tremendously far from reality. As we said before, it is mainly due to the fact that default simulation models consider only the energy induced by transmission modules, and make abstraction of the energy consumption induced by the microcontroller processing unit and the firmware. For instance, it is worth noting that the used M3 board microcontroller consumes 14 mA at full power, to which we must add the radio chip which consumes 14 mA when transmitting and 12 mA when receiving, and other energy-consuming hardware as well*. Figure 6.7 illustrates the main energy-consuming units of a smart sensor[7].

7: Note that other units can induce additional energy consumption, such as gyroscopes, accelerometers, etc.

* https://www.iot-lab.info/docs/boards/iot-lab-m3/

Table 6.9: Default and Calibrated Drawn current values for each state of the machine state used in ns-3 simulations to evaluate the power consumption of 802.15.4 communications.

State	Default Drawn Current value (mA)	Calibrated Drawn Current value (mA)
Tx	7	83
Rx	0.5	46
Tx-Busy	7	14
Rx-Busy	1.5	49
Trx-Switch	0.5	0.01
Trx-off	5×10^{-7}	5×10^{-7}

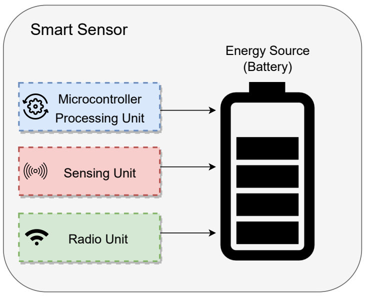


Figure 6.7: Smart Sensor Energy Consuming Units.

In order to see whether the calibrated models can be used as a baseline for the prediction of future deployments, we test the same calibrated models for a different use-case as the one used for the calibration. We consider this time that we have 60 end-devices (instead of 40), sending 2 packets (instead of 1) every second, and we calculate the energy consumption without running again our calibration. As we can see in the figure, it is still close to the reality. However, the power consumption remains practically the same when we increase the density and the traffic workload the way we did it. This may explain why the calibrated models still perform well. However, we argue that this increase is realistic in this kind of small deployment with less than 100 end-devices. Still, it would be interesting to analyze the relevance of our calibration for deployments featuring for instance more intensive traffic (*e.g.*, 200 Kbps).

As a side note, we would like to mention that the reliability observed in the simulation was also far from the real one. This suggests that, despite our efforts to replicate the deployment accurately in terms of the number and positions of the end-devices, the simulated radio environment differed significantly from the actual one. One possible explanation for this discrepancy is the presence of other active nodes in the FIT IoT-Lab platform that were in proximity to our deployed nodes. As a result, the interference caused by these additional nodes was not accounted for in the simulation, leading to differences in the reliability between the simulated and real environments.

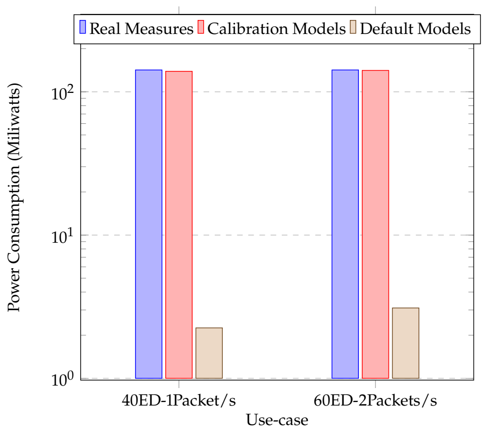


Figure 6.8: Energy Simulation Models Calibration Results.

6.3.4 Related Work

We provide here a brief overview of the related work of the energy consumption calibration. The authors of [188] provide a study of the energy-efficiency of a MAC protocol for Wireless Sensor Network (WSN)

[188]: Hurni et al. (2009), 'Calibrating Wireless Sensor Network Simulation Models with Real-world Experiments'

in simulation, and or real hardware testbed. They propose a method for cross-comparing simulation results with real-world experiments, and they show how it can reduce the gap between both. [189] propose a model for measuring the energy consumption using OMNeT++, that they calibrate using datasheets of a IEEE 802.15.4. In [190], the authors make a comparison between WSN simulators in terms of radio propagation and energy consumption. Then, they propose a calibration for the energy consumption models but with the classical approach, i.e. by measuring the current consumption for the sending and receiving. In more recent works such as [191], the authors focus on calibrating simulators in the large-scale computing platforms for Parallel and Distributed Computing (PDC) applications. They propose several accuracy metrics as well as different calibration algorithms for accuracy maximization. To the best of our knowledge, there is no work focusing on calibrating state machine models of simulation, for any network deployment, regardless of the network technology or the application.

[189]: Chen et al. (2009), 'An Energy Model for Simulation Studies of Wireless Sensor Networks using OMNeT++'

[190]: Stetsko et al. (2011), 'Calibrating and Comparing Simulators for Wireless Sensors Networks'

[191]: Koch (2021), 'An Approach for Automating the Calibration of Simulations of Parallel and Distributed Computing Systems'

6.4 Conclusion

In this chapter, we have tackled two limitations of HINTS, regarding simulation: Cost and accuracy. First, we have presented COSIMIA, a method based on simulation and machine learning to accelerate HINTS and reduce the number of simulations. The method consists of four steps: (i) Data generation using sampling, simulation and a scoring function, (ii) learning, where regression models are trained to predict the score based on the parameter combination, (iii) an inference step based on the score returned by the trained regression models and (iv) a decision step where parameter combination with the best score is retained. Regarding the learning step, according to the results of our case studies, we recommend avoiding using a linear regression model, but rather using KNN or gradient boosting. We have shown its application in the configuration optimization of an IoT technology in a given context. We tested it in three case studies including different applications and IoT technologies. Our results show that COSIMIA is able to deliver solutions close to 99% of the optimal solution, with a considerable decrease in execution time. The method has been able to propose a very good configuration (1% close to the optimal one) approximately 50 times faster than simulating all the different combinations. The main advantage of COSIMIA is that it is generic, it can be applied for any network technology, for any configuration parameters and according to any KPIs and using any simulator, as long as these aspects are available in the simulation.

Then, we have proposed a method relying on real measurements and linear regression to enhance the accuracy of the simulation models, regarding energy consumption. Our method consists of calibrating values of state machines of simulation models so that the simulated energy consumption is closer to the real one. We have validated our method using a real testbed (FIT IoT-Lab), and we saw that the default simulation models were far from the real values and that our method managed to considerably reduce this gap. Moreover, we showed that the calibrated models are able to reproduce in an accurate way a different deployment than the initial one.

In the next and final chapter, we tackle the problem of simulation complexity for IoT architects. We propose a no-code platform incorporating decision-support tools (such as HINTS) to make them accessible to IoT architects who do not necessarily have time or a deep expertise in networks and programming.

7 IoT Network Technologies No-Code Simulation

7.1 Introduction

As presented in Chapter 5, when it comes to the development of a tailored or a pre-packaged IoT solution and its deployment in a real environment, having a long-term and large-scale perspective is often critical to making the right decisions. To address the need for in-depth evaluation, network simulation appears as a key enabler. Indeed, it can provide good insights about the performance of a technology and permits to test what-if scenarios at scale to trade-off cost, QoS and energy efficiency. Network simulation complements small scale Proof of Concept (PoC) for large scale assessment and is cost efficient since there is no need to massively deploy real IoT equipment. It can be an efficient tool to help IoT solution architects evaluate the fit and future-proof the technology before setting it up in real-life scenarios. Moreover, each time the context or application is changed in production, one will have to challenge the initial assumptions and verify the actual validity of the infrastructure settings. For this, simulation is of great help all along the application lifecycle.

However, simulators like ns-3 require network expertise and knowledge in C++ programming to design experiments, to run them but also to analyze and exploit the results. Network simulators have been designed by network experts targeting network researchers and programmers, not product managers, industrial teams or even IoT architects, who do not have the time and skill to perform sophisticated simulations.

In this chapter, we present our contribution to make the network simulation power accessible and at the service of IoT teams. We propose an online no-code[1], namely StackNet, to "democratize" the use of simulation and reach a community of non-network-experts, in order to let them explore different alternatives and choices in depth. We address IoT architects to let them seamlessly benefit from network simulation via an interactive tool, for what-if scenarios creation and comparison. This enables them to future-proof the infrastructure when developing IoT applications or to test a configuration adaption during operations. This opens a path towards the incremental realization and adoption of Network Digital Twin (NDT)[2] in IoT solution design and management. For this, we have developed a model-based approach and its associated intuitive interface for setting up, running and analyzing simulations without writing a single script. Moreover, such a no-code approach for using network simulation would be an efficient way of reaching a large community of IoT researchers and professionals but also edge computing specialists and make them able to benefit from network simulation features. We believe that such a tool, potentially integrated within an NDT toolbox, can help accelerate the standardization of IoT practice to boost industry digitization and encourage contributions to open source IoT network simulators, like ns-3 towards further inclusion of more IoT network technologies.

1: It is a concept that allows users to develop software and solutions without having to write any code.

2: They are defined as virtual representations of physical objects or processes capable of collecting information from the real environment to represent, validate and simulate the physical twin's present and future behavior [192].

The remainder of this chapter is organized as follows: Problem statement is presented in Section 7.2. An overview of our approach is given in Section 7.3. Section 7.4 illustrates how our proposal can be leveraged to answer typical IoT network questions around a smart building use-case. Conclusion and future works are given in Section 7.5.

7.2 Problem Statement

The major issues that StackNet tries to solve are: (i) Evaluation an IoT network technology at scale, (ii) coding a simulation and (iii) dealing with simulation complexity. We describe each of them below:

7.2.1 Evaluating an IoT Network Technology at Scale

Selecting and configuring the network that best fits the needs of an envisioned smart application is one important and complex problem the IoT architects faces. To examine this issue, let us use a smart building example. We consider the case of the instrumentation of a commercial building already equipped with a Building Management System (BMS) in charge of the remote control of Heating, ventilation and air conditiong (HVAC) systems as well as water and energy regulation. The facility manager wants to add a smart solution to finely monitor the building, to gain better visibility on its real usage and to better adapt the building services. For this, the facility manager would like to deploy a range of sensors: Entrance detectors, occupancy monitors, air quality sensors, temperature sensors, smart lighting and other end-devices. This customer is working with IoT architects, who are proposing several sensors supporting various communication technologies. To demonstrate the feasibility of the project and to qualify the sensors for the solution, the solution architect has developed a PoC of the smart solution that shows how sensors collect data and how the prototype application exploits and visualizes them.

Asking the Right Questions

After having convinced the customer, the architects have to select and define the detailed configuration of the network together with the sensors for the targeted deployment, and integrate the final solution. They have several options for the network, typically LoRaWAN, Wi-Fi, Wi-Fi HaLow and 802.15.4 and a range of questions that the small PoC has not truly answered:

- How would LoRaWAN and Wi-Fi capacities compare for this application?
- How long will the battery last for this traffic workload?
- What are the best settings for maximizing the solution quality in this specific commercial building?
- How many devices would one gateway support?

Analyzing the Network Behavior Under Application Workload

To answer these questions, network experts would typically simulate the behavior of network candidates under application workload, considering the environmental conditions and the application traffic and topology (number of sensors, their location, etc.). They will evaluate if the traffic sent by all devices will be properly supported by the network, then estimate what percentage of packets will be successfully transmitted and how much energy will be consumed in different scenarios.

To analyze their smart building scenario in depth and at scale, the IoT architects would have to adopt the same approach: Define the application requirements as well as the network setup to be tested using a network simulator, then run a first simulation to establish the base-line and validate the parameter assumptions with one technology and a basic application scenario. Then, they would have to proceed with what-if scenarios comparison. This process requires specific skills as we detail below.

7.2.2 Coding a Simulation

Network experts generally code network experiments into the appropriated language for their simulator, for example, in C++ for ns-3. This code globally works as follows: (i) It takes input parameters, (ii) creates and executes the corresponding network nodes and traffic, (iii) calculates the KPIs obtained from the simulation.

Technology Comparison

After the base-lining step, one has to conduct technology scenario comparisons. This step consists in setting up and running a set of simulations for: (i) Various network technologies to select the best of them and (ii) different network configurations to select the best network settings for the selected network technology. The large number of technology choices and configuration parameters available could lead to the evaluation and comparison of a large set of alternatives. Having a systematic and deeper understanding of their respective behavior in terms of transmission reliability, latency or energy consumption helps to identify the weaknesses and strengths of each one for the targeted application. Network experts generally leverage their knowledge and experience to focus on the best candidates and refine gradually the analysis.

Analyzing the Scalability

After the network technology choice has been established, the scalability analysis consists in systematically stressing the selected one by the application workload to understand how an increase in load in terms of number of devices or/and traffic intensity affects the performance. The scalability generally matches the incremental way IoT deployments are made. The initial settings cover the needs for the beginning of the project. The applications and device deployments supported by the connectivity infrastructure are supposed to evolve and grow over time.

For example in the smart building case, the number of sensors is expected to increase and the topology design to change when various levels of the building will be deployed. It is then necessary to test the capacities of the network technology and each KPI under higher density or larger coverage constraints. These analyses also help to define the ideal number of gateways required to support a given load. Estimating, in advance, the KPIs for a deployment at scale or at the limits of the system capacities will ensure that the choice of the technology will survive, to some extent, the solution's scaling and the cost will not explode. Indeed, the initial design of an IoT connectivity should not limit the evolution of the applications it will support in the future. The architects should be able to deliver new services and provide enhanced performance in different ways in the coming years. Of course this flexibility has cost and interoperability requirement counterparts that simulation helps to consider very early in the project. In the case of pre-package IoT solutions, the scalability study enables vendors to adapt the technical characteristics of the solution to the needs of their market.

7.2.3 Dealing with Simulation Complexity

As we can see, this simulation workflow is far from the reach of IoT architects, who are not necessarily network experts. Consequently, they naturally tend to overlook the systematic analysis of the technology they deploy (resp. sell). This often leads to project (resp. business) failure at short, mid or long term [165].

[165]: Aramayo-Prudencio et al. (2018), 'Digital Manufacturing–Escaping Pilot Purgatory'

To change this, we propose to abstract as much as possible the complexity of the simulation process and guide IoT architects in the in-depth performance evaluation of IoT network technologies for ensuring their IoT solutions design and evolution.

7.3 Proposed Solution

In this section we explain how we transform the complex simulation process detailed above and encapsulate the simulation and analysis code described in a no-code automated network simulation software named StackNet. This software relies on the SIFRAN[3] tool, which works as a translator from ns-3 templates to forms that are executed in the background in ns-3 software. The architecture of SIFRAN is depicted in Figure 7.1.

3: SIFRAN (`https://sifran.labs.stackeo.io/`) is exclusively dedicated to the performance evaluation of IoT networks, and no decision support is included.

7.3.1 Principles

The principle of our approach is to guide the user in the methodological evaluation via a friendly user interface and to automate the coding part. This leads to the following requirements:

- Simplifying and structuring the network performance evaluation process,
- Letting the user make assumptions for the various application parameters,

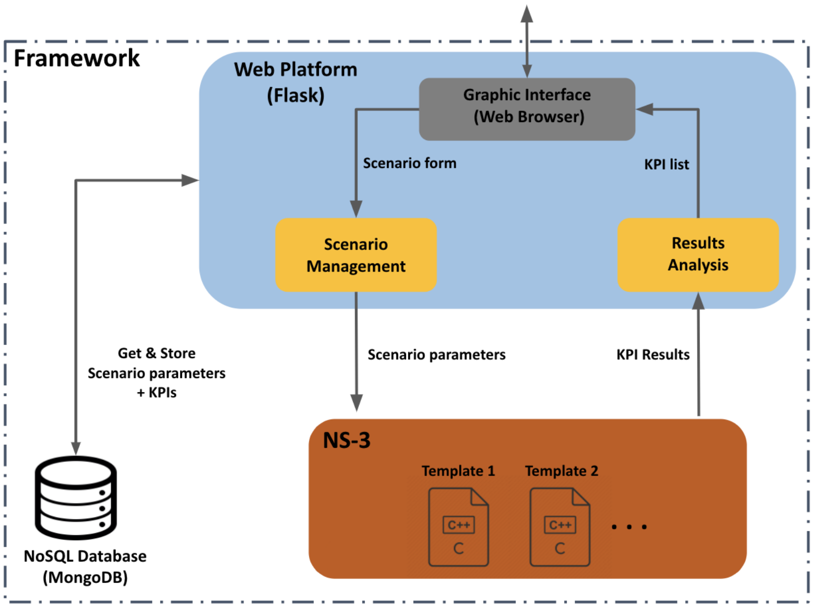


Figure 7.1: SIFRAN Architecture.

- Proposing pre-defined settings for network parameters,
- Coding automatically simulation scripts to get the KPIs evaluation via a selected simulator,
- Deploying the simulator underneath to give online access to it,
- Enabling interactive experiments,
- Gathering the results and creating easy-to-read charts,
- Making simulation data speaks for the decision-makers.

7.3.2 Methodology

To analyze the performance and the scalability of various network options with StackNet framework, IoT architects will perform the tasks listed below:

1. Identify key questions to answer for correctly supporting the targeted application.
2. Describe the application by initializing its important parameters.
3. Define a set of simulation scenarios to compare various networking alternatives.
4. Define a set of simulations to analyze the impact of the potential increase of fleet size or of workload in the future.

To get all these results, IoT architects define and run systematic experiments, changing one network parameter at a time for impact and sensibility analysis. At the end, they will be able to demonstrate the results of the integrated solution to the IoT teams, the end-customer and sales representatives. Figure 7.2 below details the steps of a comparative simulation workflow.

7.3.3 Modeling

To run each individual scenario, one has to provide the input parameters that define a scenario, and to gather the output metrics for evaluating the

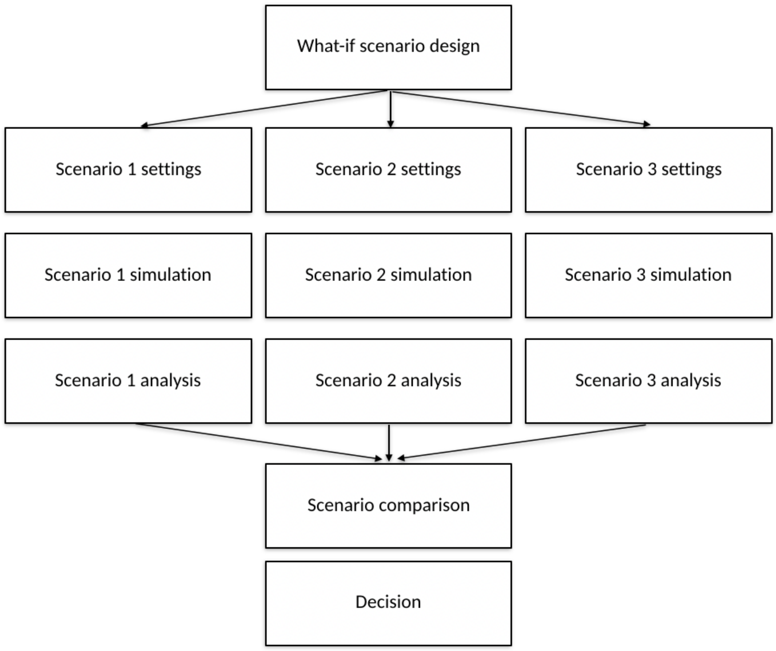


Figure 7.2: Comparative simulation Workflow with 3 Considered Scenarios.

performance. The no-code simulator abstracts what is an IoT scenario in order to be able to capture any IoT use-case and "code" it automatically in the simulator (for example ns-3) language. For this, we segregate the description of an IoT scenario in two parts: The application section and the network section. Indeed, application parameters can be easily defined by the application developer while the network settings are the difficult aspects of the simulation. The application model can be mutualized for several scenarios. We also have to propose a comprehensible way of defining the targeted output metrics (KPIs) that enable a user to analyze and compare various IoT connectivity infrastructure scenarios easily.

- **Application modeling:** For the application modeling, we rely on the framework described in Section 4.3.1.
- **Network modeling:** The specification of an IoT network simulation scenario needs a network model defined by a list of parameters representing the network technology and topology, as described in Section 4.3.2. In StackNet, pre-built network models are made available to the IoT architects, so they can integrate them in simulation scenario without network expertise.

7.3.4 Designing the No-code Interface

A simulation scenario is defined by the application model, which is set once for several experiments, and the network model and settings on the other side. Various network settings can be selected to compare what-if scenarios. To automatically generate the scripts for running a simulation, our framework provides a dynamic interface to initialize the values of the application model's parameters and the selected network settings of a scenario, as illustrated in the left column of the StackNet's interface of Figure 7.3. Pre-defined network settings with default values are proposed

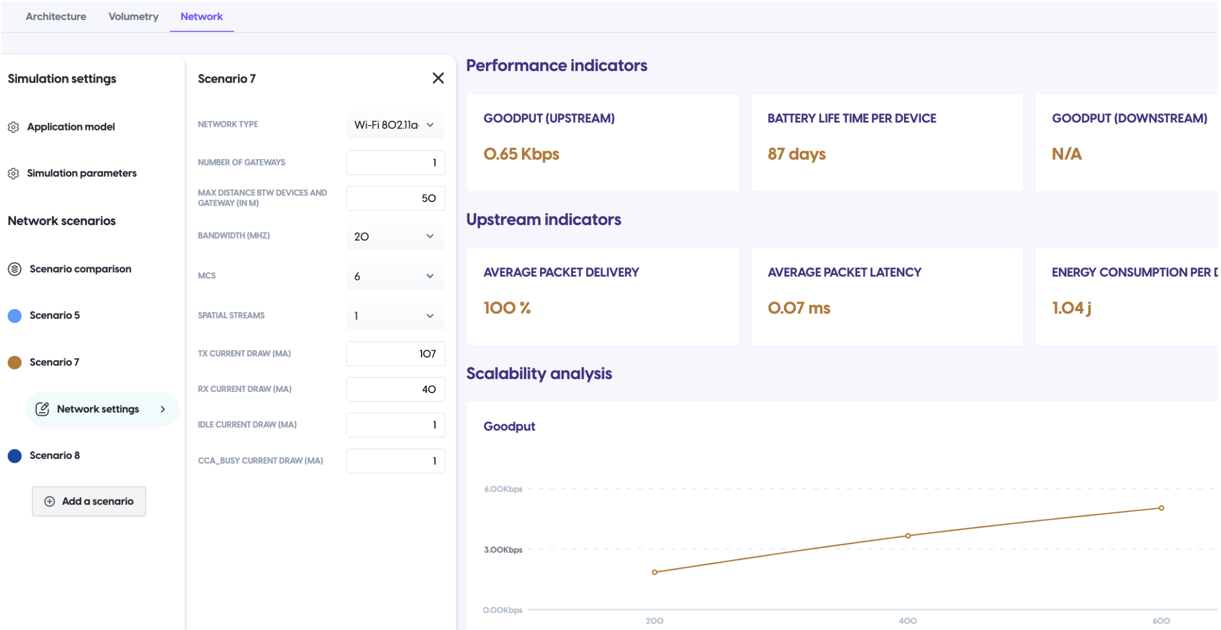


Figure 7.3: Dynamic Interface for Setting and Analyzing a Scenario.

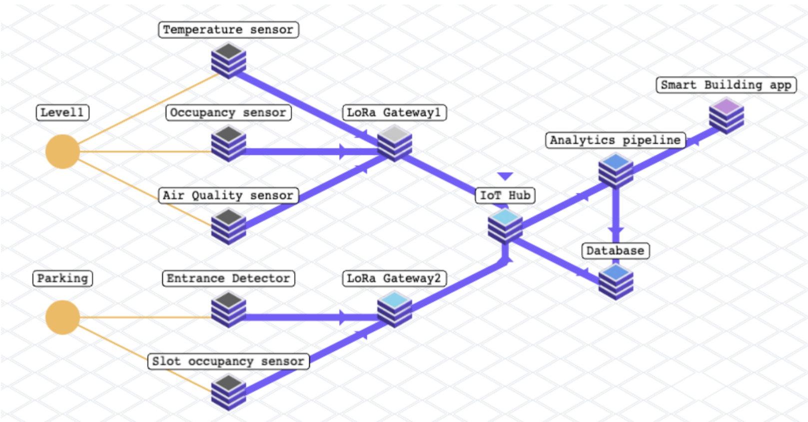


Figure 7.4: Studio of the Stackeo IoT Solution Digital Twin platform (www.stackeo.io)

for each supported technology. These pre-defined settings can be easily uploaded then adjusted manually. The interface enables to create and save multiple scenarios.

To simplify the architecture and topology design, a diagramming interface is also proposed by the Stackeo's[4] studio as illustrated in Figure 7.4. The figure represents an example for the aforementioned smart building solution, comprising: Five sensors deployed in a level and a parking, two LoRaWAN gateways, an IoT hub, a database, an analytics pipeline and finally a smart building application.

4: https://www.stackeo.io/

Once the application model and the network settings are initialized, the simulator is invoked. The simulation script is automatically generated (like described in Section 4), the simulation executed, and the resulting output metrics (KPIs) interactively visualized in scenario dashboards. To permit users to analyze and explain one particular scenario in context, they can open the input parameters panels together with the results dashboard (see Figure 7.3).

Simulation Script Generation

The steps for coding an IoT simulation scenario in an ns-3 script are described in the following:

1. **Input parameters definition**: This part of the code is where all the application and network parameters are set. These include both traffic and low-level parameters. Clearly, the considered parameters differ depending on the implemented IoT network technology.
2. **Nodes placement**: This code section creates all the nodes (end-devices and gateways) and places them in three dimensional space.
3. **Layers configuration**: The network technology is defined here by setting its physical, mac and network layers.
4. **Low-level parameters configuration**: The low-level parameters which have been declared such as the spreading factor for LoRaWAN are instantiated and set at the nodes level here.
5. **IP address configuration**: In case the IP addresses are supported in the nodes, we configure them in this part in order to make the nodes accessible to each other.
6. **Application traffic specification**: This part is where the traffic definition is made. Depending on the traffic type, applications specification (packet size, etc.) are defined and installed in the nodes, with fixing the destination address.
7. **Energy configuration**: To keep trace of the consumed energy during the simulation, an energy source and a draining model are configured on nodes. The energy source can either be linear or nonlinear. The latter considers the inherent battery discharge and recharge [193]. The energy consumption models are based on state-machines, which assign to each physical state a current draw consumption in milliamperes.
8. **Trace files generation**: There is the possibility in ns-3 of generating Packet Capture (pcap) and tracing files which contain all the packets that have flowed through the network. It is worth noting that pcap files can be opened with software like Wireshark, while the trace files can be read using any text editor.
9. **KPIs calculation**: At the end of the template, all the targeted KPIs (packet throughput, packet latency, reliability, energy consumption and battery lifetime) extracted from simulation are gathered.

[193]: Rakhmatov et al. (2001), 'An Analytical High-level Battery Model for Use in Energy Management of Portable Electronic Systems'

The results dashboard gathers and highlights the KPIs related to the IoT connectivity solution under study in the specified application context as illustrated in the Figure 7.4. The network performance evaluation focuses on five KPIs. These KPIs are: (i) Goodput (or throughput), (ii) reliability, (iii) packet latency, (iv) energy consumption and (v) battery lifetime. We consider that, together, these metrics provide a fair representation of the performance of an IoT network technology for a given scenario.

If the application requires high quality data, like for precision air quality monitoring in the operating room of a hospital, then, estimating the reliability is of utmost importance. For a use-case like real-time equipment location application, packet latency is what matters the most. If the deployment of sensors and the battery change are difficult, energy consumption and battery life time have critical importance.

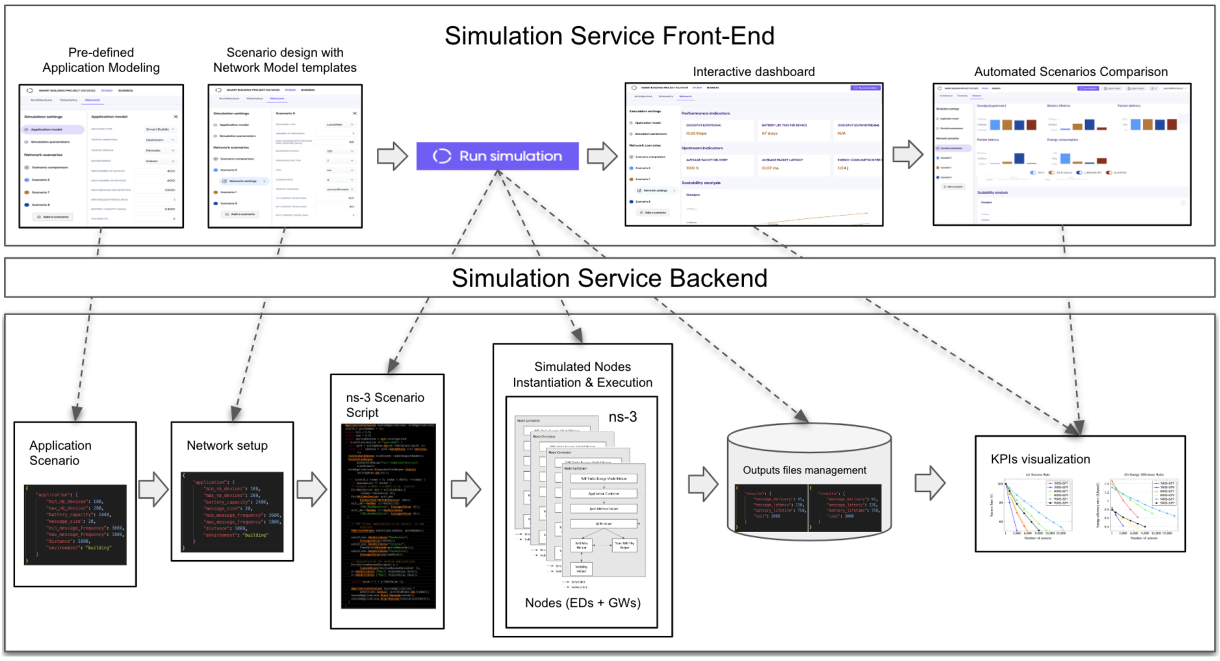


Figure 7.5: Illustration of the No-code Workflow and its Mapping to Expert simulation for IoT Scenarios Analysis.

As the IoT architects are looking for scenario comparison, a dedicated and intuitive dashboard (*e.g.,* Figure 7.7) is proposed to display comparative results and charts, and to let the best technology and configuration immediately stand out.

The evaluation methodology has been seamlessly integrated within the interactive front-end while the programming and networking expertise have been encoded and hidden within the back-end. This makes the simulation experience smooth and quick, allowing the IoT architects to focus on the data and decision process rather than bothering them with programming and results collection complexity. Figure 7.5 illustrates the no-code workflow for creating and setting various scenarios for comparison, and the corresponding mapping to the simulation workflow activated underneath.

7.4 Use-case Example

This section details how IoT architects would use StackNet for the aforementioned smart building use-case. The application model is defined by the following: 100 end-devices are placed in a building. They are separated by a distance of around 200 meters. They send one packet of 100 bytes every 2 minutes. For the scalability analysis, the network density is scaled (up to 600 end-devices) and the traffic workload is increased (one packet of 110 bytes is sent every 90 seconds). The IoT architects introduce the application model inputs, which can be found in Table 7.1. Figure 7.6a displays the Application model inputs as entered in the online tool.

Then, IoT architects create various scenarios for different network technologies, typically Wi-Fi, 802.15.4, 802.11ah and LoRaWAN. Figure 7.6b

Table 7.1: Application Modeling of Case Study.

Application modeling	Parameters	Case Study
End-devices	• Minimal number • Maximal number • Battery capacity (Amperes.hour)	100 600 2.4
Workload	• Traffic direction • Message size (bytes) • Minimal frequency (packets/second) • Maximal frequency (packets/second)	Upstream {100, 110} 0.016 0.032
Environment	• Type • Scope (meters) • Expected lifetime (days)	Indoor 200 N/A

gives the inputs for a LoRaWAN scenario generation.

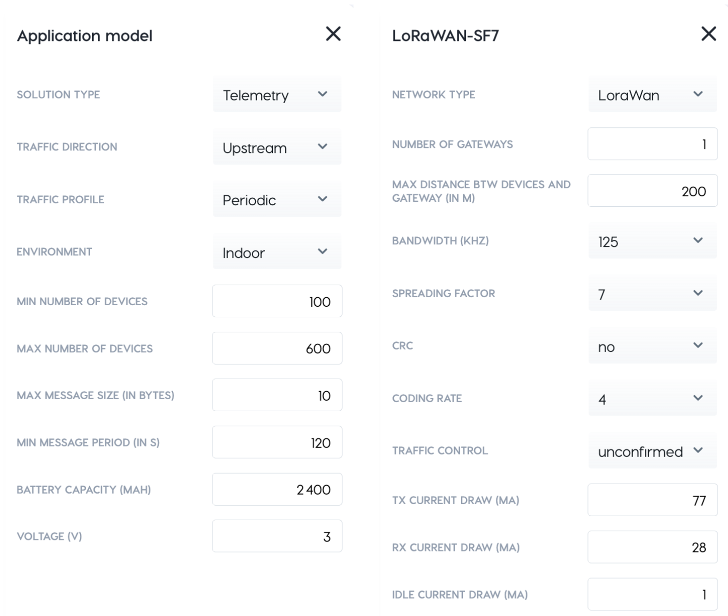
Let us consider how IoT architects can leverage the network simulation service to easily answer the questions asked in Section 7.2:

1) How would LoRaWAN and Wi-Fi capacities compare for this smart building application? Figure 7.7 displays the performance summary of the different technologies explored. We can easily observe that all the technologies perform the same in terms of goodput and reliability, *i.e.* all the packets are correctly received. However, LoRaWAN clearly outclasses the other technologies, including Wi-Fi, in terms of battery lifetime (up to 700 days). Therefore, we can say that LoRaWAN is more relevant than Wi-Fi in the basic version of this use-case.

2) How long will the battery last? The battery lifetime of Wi-Fi can last approximately 3 months (90 days) while 802.15.4 can go up to 5 months (150 days). The difference is not that big, but it shows that 802.15.4 is better designed and suited for this IoT applications than Wi-Fi. For LoRaWAN, it can go up to approximately 2 years.

3) What are the best settings with the LoRaWAN technology (spreading factor) to maximize the solution quality in this specific commercial building? Figure 7.8 shows how the IoT architects can further explore some configurations of LoRaWAN with different spreading factors (7, 8 and 9) to determine the impact of this parameter on the performance. Except covering longer distances, we observe that higher Spreading Factor (SF) is less interesting in terms of all the considered Key Performance Indicators (KPI). Packet latency is increased, which makes collisions more likely, and therefore lessens reliability. Additionally, end-devices spend more time to send a packet, and thus consume more energy, leading to lower battery lifetime. We see that a lower SF results in lower message latency, better reliability and longer battery lifetime. In this case SF of 7 is the best setting.

4) How many devices would one gateway support? Figures 7.9a and 7.9b highlight the various network technologies' performance in terms of reliability and packet latency for the scaled settings. We see that Wi-Fi HaLow, 802.15.4 and Wi-Fi behave well in terms of reliability. However,



(a) Interface for Modeling the Application. **(b)** Interface for Network Settings.

Figure 7.6: Application Model and Network Settings Abstraction.

it drops faster for LoRaWAN (down to 60% for 600 end-devices). The scalability study shows that one LoRaWAN gateway can handle up to 200 sensors before deteriorating the performances, while one Wi-Fi HaLow gateway can handle 400 sensors without noticing any degradation in the performances. The node's placement and the radio environment in this use-case make it impossible to use only one gateway for Wi-Fi and 802.15.4.

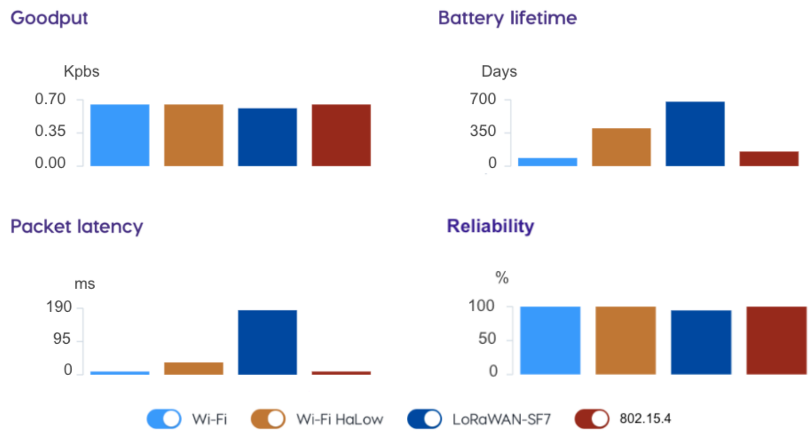


Figure 7.7: Results for the Network Technologies Comparison (Q1 & Q2).

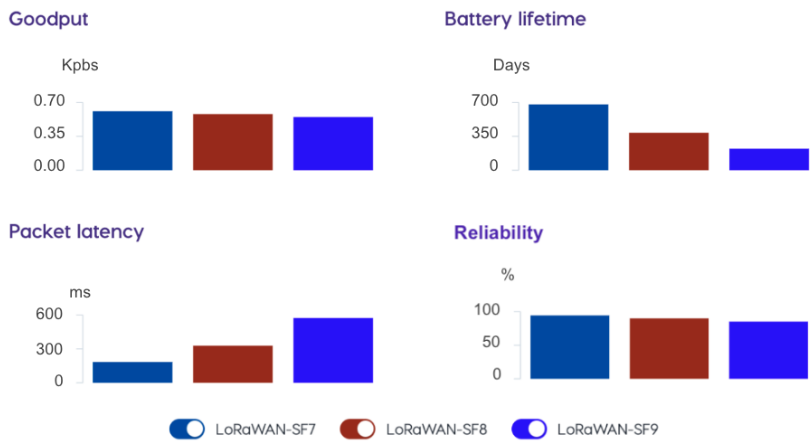
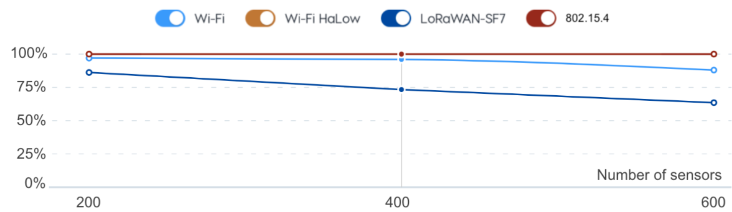
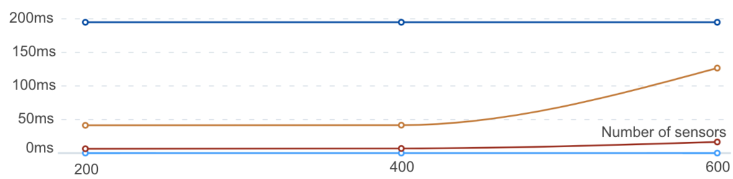


Figure 7.8: Results for the LoRaWAN Configuration (Q3).



(a) Reliability.



(b) Packet Latency.

Figure 7.9: Results for the Scalability Study (Q4).

7.5 Conclusion

In this chapter, we have presented StackNet, a no-code framework to ease and automate IoT network simulation and its integration in a SaaS (Software as a Service) platform. This approach enables IoT architects to test and compare IoT network technologies without learning and deploying any network simulator nor coding any script. We began by identifying the network evaluation problems that IoT architects face on their journey, and the difficulty of running network simulations. We described our approach by highlighting the salient aspects that need to be taken into consideration for hiding the complexity of IoT simulation while empowering architect with an intuitive solution to design and compare alternative scenarios.

An application of our methodology and tool on a smart building use-case has been presented. We put the emphasis on the ease of simulation initialization, on results visualization, on what-if scenarios comparison. We show how the no-code framework helps IoT architects ask and answer their own design questions. The tool currently supports simulations with Wi-Fi, LoRaWAN, Wi-Fi HaLow, 802.15.4 and 5G mmWave, but is limited by the availability of network technologies in ns-3. As the no-code and as a service approach can democratize the usage of cost-effective evaluation methods like simulation in IoT, we hope that our no-code tool dedicated to the simulation of IoT network technologies will encourage the systematic development of ns-3 simulation modules for other IoT network technologies. StackNet can be freely accessed via the Stackeo digital twin platform here `https://app.stackeo.io/`. We believe that such a tool, potentially integrated within a network digital twin toolbox, can help accelerate the standardization of IoT practice to boost industry digitization and encourage contributions to open source IoT network simulators.

This study was conducted in collaboration with Stackeo[5], a company that provides collaborative software and consulting services to support businesses and integrators in planning, scaling, and optimizing smart connected solutions efficiently and quickly. Stackeo enhances IoT teams by incorporating Modeling, Simulation, and Digital Twin Network technology, enabling accelerated progress and ensuring the profitability, reliability, and sustainability of strategic projects.

5: `https://stackeo.io/`

Conclusion & Perspectives | 8

The main goal of this thesis was to propose solutions that allow IoT architects to make informed decisions during the design and ongoing management phases of their IoT solution lifecycle. In particular, we were interested in the problems of the selection and configuration of IoT network technologies for a given IoT solution. In this chapter, we provide a summary of the major contributions of this thesis. Then, we give their major limitations and we finally give some perspectives and future works.

8.1 Contributions

8.1.1 Modeling and Evaluation

We have introduced a framework that assesses the performance of network technologies in IoT applications, where multiple end-devices communicate through gateways. The framework includes defining a scenario and its key performance indicators (KPIs) for evaluation. To demonstrate its applicability, we applied the framework to two use cases inspired by real-life IoT applications, examining different network technologies. We particularly focused on energy efficiency and scalability with increasing end-devices. The evaluation results, based on the specific applications, emphasize the significance of considering a comprehensive approach to evaluate the suitability of communication technologies in their specific contexts.

8.1.2 Selection

We have presented HINTS, a methodology designed to support the selection and configuration of IoT network technologies. HINTS utilizes a combination of IoT network technology modeling and a five-step decision process. The chapter outlines each step of the process, which includes: (i) Application modeling, where the IoT application's specific requirements and key performance indicators (KPIs) are abstracted; (ii) Pre-selection, which eliminates unsuitable network technologies from consideration; (iii) Scenario design, where the application is configured with potential network technology candidates; (iv) Evaluation, involving iterative identification of the best-suited topology using techniques such as simulation to estimate KPIs in the targeted application scenario; and (v) Decision-making, where scores are assigned to each alternative using a Multi Attibute Decision Making (MADM) method derived from Technique for Order Preference Similarity to Ideal Solution (TOPSIS). The results demonstrated that HINTS enables a comprehensive and informative comparison of IoT network technologies within a given application scenario. Additionally, it facilitates the exploration and determination of network configuration parameters and the number of gateways required for deployment. The study emphasizes the significance of considering the application context, environment, and scaling factor in the network selection process and expected performance.

8.1.3 Limits of Simulation

Simulation Time

We have introduced COSIMIA, a methodology combining simulation and machine learning that accelerates the design decision of HINTS by reducing the number of simulations. We showed an application on network configuration optimization. COSIMIA consists of three main steps: (i) data generation, which involves sampling, simulation, and a scoring function to generate data points for training; (ii) learning, where regression models are trained to predict scores based on the network configuration; and (iii) inference step, which utilizes the trained regression models to search for the optimal configuration based on the predicted scores. The results indicated that COSIMIA could obtain promising results on a couple of different examples, featuring different IoT applications with different network technologies.

Simulation Reliability

We explored how experimentation can be coupled with simulation to make the latter more grounded. Precisely, we proposed a method to calibrate the energy consumption of the models used by the simulation to make them closer to reality. We introduced a method that utilizes linear regression to calibrate the current consumption values of state machines in order to make the simulated energy consumption more accurate. We conducted experiments using a real testbed (FIT IoT-Lab) and observed significant discrepancies between the default models and the actual values. Our proposed method successfully reduced this gap, improving the realism of the simulation. We showed how our method could be used with a small scale Proof of Concept (PoC) during the design phase, where there is access to energy measures.

8.1.4 No-code Simulation

We presented StackNet, a framework designed to simplify and automate IoT network simulation and its integration into a Software as a Service (SaaS) platform, without requiring users to learn complex network simulators or write code. We address the challenges faced by IoT architects in evaluating IoT network technologies and the difficulties associated with running network simulations. Our approach focused on providing an intuitive solution that allows architects to design and compare alternative scenarios while hiding the complexity of IoT simulation. We outline the key aspects of our methodology, which aims to empower architects by facilitating simulation initialization, results visualization, and comparison of what-if scenarios. The application of StackNet emphasized the ease of use in simulation initialization, visualizing results, and comparing different scenarios.

8.2 Limitations & Perspectives

8.2.1 Modeling and Evaluation

First, one of the main limitations of our framework is that the considered energy consumption is only limited to transmission. Sensing and processing may induce additional energy consumption that is not captured by this framework. Also, the environmental impact should be considered by taking into account all the chain (creation of end-devices, recycling, etc.). Then, since our framework mainly relies on simulation, it is highly tied to the availability of the network technologies in the used simulator. Due to the requirements of abstraction and generalization of the study, we saw simulation as a natural choice, and indeed it allowed us to make the comparison of several network technologies using the same tool, namely ns-3. However, it was a complex task to integrate network technologies and their energy models that were not part of the official release of ns-3 but distributed through various unofficial patches. We believe that having a unified framework for simulating different network technologies on the same simulator can be of great interest to the research community. Moreover, Low Power Wide Area Network (LPWAN) technologies like NB-IoT, LTE-M, Sigfox, or even satellite communication still have to be included in our framework (a recent work [194] has proposed a module for simulating Sigfox networks in ns-3). In this context, heterogeneous networks, which involve the integration and interconnection of different network technologies within a single IoT deployment, become particularly relevant. They enable the seamless coexistence and interaction of various communication technologies, such as cellular networks, Wi-Fi, Bluetooth, Zigbee, and satellite communication, among others. The inclusion of heterogeneous networks in our framework could allow us to address the challenges and opportunities associated with diverse technologies, such as the dynamic network selection during the operation phase of the IoT solution.

[194]: Naeem et al. (2022), 'A Sigfox Module for the Network Simulator 3'

Another major limitation of our work is that our modeling does not consider mobile use-cases. Applications like asset tracking, fleet management, location-based services and diverse smart city applications are gaining a lot of interest. It would be interesting to explore how simulation can be used to help in the design of this kind of application. In fact, we have tested some mobile use-cases with Wi-Fi HaLow and LoRaWAN in ns-3[1], but it resulted in strange results where the end-to-end reliability was extremely low. It would be interesting to investigate this issue and see how it can be solved. Finally, mesh networks and hierarchical network interconnections, utilizing routing protocols like Routing Protocol for Low-Power and Lossy Networks (RPL), offer alternative architectures for IoT deployments that were not investigated in this thesis. These alternative architectures introduce additional decision considerations, such as optimizing power-efficient routing and load-balancing strategies. While these aspects are not addressed in this thesis, they present interesting avenues for further exploration in the realm of IoT network design and optimization. Finally, it is worth recalling that this thesis only tackles issues related to the PHY and MAC of IoT solutions. The upper layers of the IoT architecture, including the application and service layers, face several challenges. These challenges, which include efficient

1: It is possible to simulate mobile end-devices in ns-3 using different Mobility Models. Details can be found here: `https://www.nsnam.org/docs/models/html/mobility.html`.

data management and processing, interoperability and standardization, security and privacy concerns, also need a thorough investigation.

8.2.2 Selection

The major limitation of HINTS concerns the subjectivity in the use of TOPSIS (and MADM methods in general). On the one hand, we need to keep in mind that MADM methods are just a way of scoring alternatives according to given criteria: The scores they provide are absolutely not to be considered as ground truth for the performance comparison. On the other hand, other more sophisticated methods for the multi-criteria optimization can be considered, such as [195] where one can choose to explore different regions of Pareto front[2]. This means that some criteria may be favored compared to others in a more subtle way than just using multiplicative weights, as done in MADM methods.

[195]: Paria et al. (2020), 'A Flexible Framework for Multi-objective Bayesian Optimization using Random Scalarizations'

2: It is a concept used in multi-objective optimization to describe the set of solutions that cannot be improved in one objective without worsening at least one of the other objectives [196].

8.2.3 Limits of Simulation

Simulation Time

A natural limitation of COSIMIA is that it works like a heuristic. Indeed, even though it seems to work well on the presented use-cases, there is no guarantee that the method will manage to find a solution near the optimal. Moreover, the proposed sampling (min-mid-max) for the data generation step may let us think that the method works on parameters that have a monotone influence on the score provided by TOPSIS. If it were not the case, the sampling would miss some important values and thus would probably generate worse results that the presented ones. For that sake, we have initiated work with a colleague where we propose a method for optimizing the network configuration using Bayesian optimization, which allows a more robust way of exploring the data to find the optimal configuration. Bayesian learning or optimization is a type of statistical learning that uses Bayesian probability theory to make predictions and decisions based on acquired data[3]. Maintaining a probability distribution over the solution space enables systematic exploration of different regions while exploiting areas that are likely to yield better solutions. Indeed, the obtained results are very promising, and the solution is able to find optimized configurations in even less time than COSIMIA.

3: It is called Online Learning, since the learning process is done gradually and not on an established dataset, as it is the case for other learning algorithms.

Simulation Reliability

The main limiting factor of the presented calibration method is the fact that it only works for a given deployment. Indeed, energy consumption is strongly tied to the type of the end-device, and affected by the deployment, the radio conditions, etc. Thus, the calibrated simulation models are not intended to be used on other deployments with other end-devices. Moreover, our method assumes the availability of periodic energy consumption measures from the PoC. This assumption may be relatively strong and restrictive. It is indeed challenging to offer a measuring platform for the energy consumption of the end-devices. This would mean that there are typically watt meters connected to the end-devices. It can be very complicated to provide such a measuring platform. One way of avoiding these restrictive assumptions would be to have benchmarks of energy consumption of different application models (as defined in Chapter 4), classified according to their characteristics (density, workload, etc.). Then, to calibrate a simulated deployment with a given application, one could use data from an application of the same class. As a perspective, it would be interesting to see how our method can enhance classical Network Digital Twin (NDT) architectures, which hold a permanent connection between the virtual and the physical twin, *i.e.,* the physical network, along the whole solution lifecycle. This way, a calibration can be triggered every time the deployment changes (in terms of density or traffic workload). However, it would be of interest to analyze the cost of such a permanent connection and its advantages for the design process, compared to a small-scale PoC during the design phase only, as we did.

Moreover, it would be interesting to see if we can calibrate other KPIs such as the radio link quality for instance. To accurately reproduce a real radio environment in simulation is still an open challenge.

8.2.4 No-code Simulation

Regarding the limitations of StackNet, as the user interface is simulator-agnostic, we plan to adapt the back-end to support other simulators like OMNeT++ and even experimental test-beds to leverage the same user-friendly tool for technology and configuration comparison. In terms of future works, we will also extend StackNet with a decision support engine to generate recommendations for network selection and configuration. We will also combine network simulation with holistic and global IoT solution monitoring for precise estimation of the financial cost and the environmental impact analysis.

8.3 Concluding Remarks

All these different perspectives show that the contributions of this thesis still have a large room for improvement, and there is still a lot to do. However, we believe that the results obtained through this work confirm the significance of this research area, and we hope that they will serve the community of researchers and of IoT architects, fostering stronger relationships and collaboration between them. By sharing a common objective of enhancing connectivity and services for a wide user base, this collaboration can lead to meaningful advancements and better meet the needs of the IoT ecosystem.

From a personal perspective, this research work has made me realize the complexity and challenges of the IoT field, such as network simulation, mastering network technologies, or even abstraction, formalization and generalization of IoT applications. Network simulation, in particular, can be a daunting task, requiring a deep understanding of various simulation tools, models, and protocols. The process of mastering network technologies is equally complex. Furthermore, formalizing IoT applications poses its own set of difficulties, as it demands a comprehensive understanding of real-world scenarios and the ability to capture user needs in a formal framework.

Despite the inherent challenges, my conviction is that the perspectives are very promising. Delving into these areas was an exciting experience that I have greatly appreciated. Navigating through the intricacies of network simulation and mastering network technologies has allowed me to significantly expand my knowledge and skills, and has provided me with valuable insights into the inner workings of networking systems. Finally, I hope that the results presented in this thesis may be reused in further research work, enabling me to contribute to the advancement of the field.

Bibliography

Here are the references in citation order.

[1] Philippa Biggs et al. 'Harnessing the Internet of Things for Global Development'. In: *International Telecommunication Union, Cisco* (2016) (cited on page 1).

[2] Lantronix Inc. 'Product Life Cycle in the Age of IoT'. In: (2020) (cited on page 2).

[3] Rahul Badnakhe. 'A Blueprint on IoT Solutions Development'. In: *eInfochips* (2022) (cited on page 2).

[4] Itorobong S Udoh and Gerald Kotonya. 'Developing IoT Applications: Challenges and Frameworks'. In: *IET Cyber-Physical Systems: Theory & Applications* 3.2 (2018), pp. 65–72 (cited on page 3).

[5] *Workforce of the Future to Benefit from New Digital Technologies*. URL: https://newsroom.accenture.com/subjects/management-consulting/industrial-internet-of-things-will-boost-economic-growth-but-greater-government-and-business-action-needed-to-fulfill-its-potential-finds-accenture.htm (cited on page 3).

[6] M Chui, M Collins, and M Patel. 'The Internet of Things: Catching up to an Accelerating Opportunity'. In: *McKinsey & Company, New York* (2021) (cited on page 3).

[7] ABI research. 'IoT Asset Tracking Device Evolution: Technologies and Features Driving the Next Stage of Growth'. In: *ABI research* (2020) (cited on page 3).

[8] Miao Wu et al. 'Research on the Architecture of Internet of Things'. In: *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*. Vol. 5. IEEE. 2010, pp. V5–484 (cited on page 4).

[9] In Lee and Kyoochun Lee. 'The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises'. In: *Business horizons* 58.4 (2015), pp. 431–440 (cited on page 4).

[10] Wafa'a Kassab and Khalid A Darabkh. 'A–Z survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations'. In: *Journal of Network and Computer Applications* 163 (2020), p. 102663 (cited on pages 4, 29, 34).

[11] Emmanuel Baccelli. 'Internet of Things (IoT): Societal Challenges & Scientific Research Fields for IoT'. In: (2021) (cited on page 4).

[12] Frederic Vannieuwenborg, Sofie Verbrugge, and Didier Colle. 'Choosing IoT-connectivity? A guiding Methodology based on Functional Characteristics and Economic Considerations'. In: *Transactions on emerging telecommunications technologies* 29.5 (2018) (cited on pages 5, 42, 81, 82).

[13] Heidelberger and Lavenberg. 'Computer Performance Evaluation Methodology'. In: *IEEE Transactions on Computers* 100.12 (1984), pp. 1195–1220 (cited on page 7).

[14] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann, 2014 (cited on page 7).

[15] Robert E Shannon. 'Introduction to the Art and Science of Simulation'. In: *1998 winter simulation conference. proceedings (cat. no. 98ch36274)*. Vol. 1. IEEE. 1998, pp. 7–14 (cited on page 7).

[16] Shahadat Uddin, Stephen Ong, and Haohui Lu. 'Machine Learning in Project Analytics: A Data-driven Framework and Case Study'. In: *Scientific Reports* 12.1 (2022), p. 15252 (cited on page 7).

[17] Samir Si-Mohammed et al. 'HINTS: A Methodology for IoT Network Technology and Configuration Decision'. In: *Internet of Things* (2023), p. 100678 (cited on page 10).

[18] Samir Si-Mohammed et al. 'ADIperf: A Framework for Application-driven IoT Network Performance Evaluation'. In: *2022 International Conference on Computer Communications and Networks (ICCCN)*. IEEE. 2022, pp. 1–8 (cited on pages 11, 81).

[19] Samir Si-Mohammed et al. 'SIFRAN: Evaluating IoT Networks with a No-code Framework based on ns-3'. In: *Proceedings of the 2022 Latin America Networking Conference*. 2022, pp. 42–49 (cited on page 11).

[20] Samir Si-Mohammed et al. 'StackNet: IoT Network Simulation as a Service'. In: *IEEE International Conference on Communications (ICC 2023)*. 2023 (cited on page 11).

[21] Samir Si-Mohammed et al. 'COSIMIA: Combiner Simulation et Apprentissage Automatique pour Optimiser la Configuration des Réseaux IoT'. In: *CoRes 2023-8èmes Rencontres Francophones sur la Conception de protocoles, l'évaluation de performances et l'expérimentation de Réseaux de communication*. 2023 (cited on page 11).

[22] Samir Si-Mohammed et al. 'Supporting Unmanned Aerial Vehicle Services in 5G Networks: New High-level Architecture integrating 5G with U-space'. In: *IEEE Vehicular Technology Magazine* 16.1 (2020), pp. 57–65 (cited on page 11).

[23] Samir Si-Mohammed et al. 'UAV Mission Optimization in 5G: On Reducing MEC Service Relocation'. In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6 (cited on page 11).

[24] IoT Analytics. *Cellular IoT & LPWA Connectivity Market Tracker (2020)*. 2022 (cited on page 13).

[25] Neha Sharma, Madhavi Shamkuwar, and Inderjit Singh. 'The History, Present and Future with IoT'. In: *Internet of things and big data analytics for smart generation* (2019), pp. 27–51 (cited on page 14).

[26] John H Manley. 'Embedded Computers: Software Cost Considerations'. In: *Proceedings of the May 6-10, 1974, national computer conference and exposition*. 1974, pp. 343–347 (cited on page 14).

[27] Kevin Ashton et al. 'That 'Internet of Things' thing'. In: *RFID journal* 22.7 (2009), pp. 97–114 (cited on page 14).

[28] Somayya Madakam et al. 'Internet of Things (IoT): A Literature review'. In: *Journal of Computer and Communications* 3.05 (2015), p. 164 (cited on page 14).

[29] Thomas Watteyne and Kristofer SJ Pister. 'Smarter Cities through Standards-based Wireless Sensor Networks'. In: *IBM Journal of Research and Development* 55.1.2 (2011), pp. 7–1 (cited on page 15).

[30] Renato Caminha Juacaba Neto et al. 'Scalability of LPWAN for Smart City Applications'. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2021, pp. 74–79 (cited on page 15).

[31] Steven Latre et al. 'City of Things: An Integrated and Multi-technology Testbed for IoT Smart City Experiments'. In: *2016 IEEE international smart cities conference (ISC2)*. IEEE. 2016, pp. 1–8 (cited on pages 15, 31).

[32] Noelia Uribe-Pérez and Carles Pous. 'A Novel Communication System Approach for a Smart City based on the Human Nervous System'. In: *Future Generation Computer Systems* 76 (2017), pp. 314–328 (cited on page 15).

[33] Vasileios A Memos et al. 'An Afficient Algorithm for Media-based Surveillance System (EAMSuS) in IoT Smart City Framework'. In: *Future Generation Computer Systems* 83 (2018), pp. 619–628 (cited on page 15).

[34] Moeen Hassanalieragh et al. 'Health Monitoring and Management using Internet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges'. In: *2015 IEEE international conference on services computing*. IEEE. 2015, pp. 285–292 (cited on page 15).

[35] Arijit Ukil et al. 'IoT Healthcare Analytics: The Importance of Anomaly Detection'. In: *2016 IEEE 30th international conference on advanced information networking and applications (AINA)*. IEEE. 2016, pp. 994–997 (cited on page 15).

[36] Sara Amendola et al. 'RFID Technology for IoT-based Personal Healthcare in Smart Spaces'. In: *IEEE Internet of things journal* 1.2 (2014), pp. 144–152 (cited on page 15).

[37] Priyan Malarvizhi Kumar and Usha Devi Gandhi. 'A Novel Three-tier Internet of Things Architecture with Machine Learning Algorithm for Early Detection of Heart Diseases'. In: *Computers & Electrical Engineering* 65 (2018), pp. 222–235 (cited on page 15).

[38] Tien Cao-Hoang and Can Nguyen Duy. 'Environment Monitoring System for Agricultural Application based on Wireless Sensor Network'. In: *2017 Seventh International Conference on Information Science and Technology (ICIST)*. IEEE. 2017, pp. 99–102 (cited on page 16).

[39] Ji-chun Zhao et al. 'The Study and Application of the IoT Technology in Agriculture'. In: *2010 3rd international conference on computer science and information technology*. Vol. 2. IEEE. 2010, pp. 462–465 (cited on page 16).

[40] Zhao Liqiang et al. 'A Crop Monitoring System based on Wireless Sensor Network'. In: *Procedia Environmental Sciences* 11 (2011), pp. 558–565 (cited on page 16).

[41] Duan Yan-e. 'Design of Intelligent Agriculture Management Information System based on IoT'. In: *2011 Fourth International Conference on Intelligent Computation Technology and Automation*. Vol. 1. IEEE. 2011, pp. 1045–1049 (cited on page 16).

[42] Tamoghna Ojha, Sudip Misra, and Narendra Singh Raghuwanshi. 'Wireless Sensor Networks for Agriculture: The State-of-the-Art in Practice and Future Challenges'. In: *Computers and electronics in agriculture* 118 (2015), pp. 66–84 (cited on page 16).

[43] Mussab Alaa et al. 'A Review of Smart Home Applications based on Internet of Things'. In: *Journal of Network and Computer Applications* 97 (2017), pp. 48–65 (cited on page 16).

[44] Shiu Kumar. 'Ubiquitous Smart Home System using Android Application'. In: *International Journal of Computer Networks & Communications* 6 (2014) (cited on page 16).

[45] Akram Khan et al. 'Design of an IoT Smart Home System'. In: *2018 15th Learning and Technology Conference (L&T)*. IEEE. 2018, pp. 1–5 (cited on page 16).

[46] Timothy Malche and Priti Maheshwary. 'Internet of Things (IoT) for Building Smart Home System'. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE. 2017, pp. 65–70 (cited on page 16).

[47] Hui Yang et al. 'The Internet of Things for Smart Manufacturing: A Review'. In: *IISE transactions* 51.11 (2019), pp. 1190–1216 (cited on page 17).

[48] Jian Qin, Ying Liu, and Roger Grosvenor. 'A Framework of Energy Consumption Modelling for Additive Manufacturing using Internet of Things'. In: *Procedia CIRP* 63 (2017), pp. 307–312 (cited on page 17).

[49] Yee Shee Tan, Yen Ting Ng, and Jonathan Sze Choong Low. 'Internet-of-Things enabled Real-time Monitoring of Energy Efficiency on Manufacturing Shop Floors'. In: *Procedia CIRP* 61 (2017), pp. 376–381 (cited on page 17).

[50] Yingfeng Zhang et al. 'Real-time Information Capturing and Integration Framework of the Internet of Manufacturing Things'. In: *International Journal of Computer Integrated Manufacturing* 28.8 (2015), pp. 811–822 (cited on page 17).

[51] Pascale Vicat-blanc. *Systems and Methods for Modeling and Simulating an IoT System*. US Patent App. 17/515,911. 2022 (cited on pages 17, 18).

[52] John A Stankovic and Raj Rajkumar. 'Real-time Operating Systems'. In: *Real-Time Systems* 28.2-3 (2004), pp. 237–253 (cited on page 18).

[53] Emmanuel Baccelli et al. 'RIOT OS: Towards an OS for the Internet of Things'. In: *2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE. 2013, pp. 79–80 (cited on pages 18, 97).

[54] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. 'Contiki-a Lightweight and Flexible Operating System for Tiny Networked Sensors'. In: *29th annual IEEE international conference on local computer networks*. IEEE. 2004, pp. 455–462 (cited on pages 18, 34).

[55] Adam Dunkels, Oliver Schmidt, and Thiemo Voigt. 'Using Protothreads for Sensor Node Programming'. In: *Proceedings of the REALWSN*. Vol. 5. Citeseer. 2005 (cited on page 18).

[56] Richard Barry et al. 'FreeRTOS'. In: *Internet, Oct* 4 (2008) (cited on page 18).

[57] Zephyr. *Zephyr Project Documentation*. 2005. URL: %7Bhttps://docs.zephyrproject.org/latest/introduction/index.html%7D (cited on page 18).

[58] Hubert Zimmermann. 'OSI Reference Model-the ISO Model of Architecture for Open Systems Interconnection'. In: *IEEE Transactions on communications* 28.4 (1980), pp. 425–432 (cited on page 19).

[59] Kais Mekki et al. 'Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT'. In: *2018 ieee international conference on pervasive computing and communications workshops (percom workshops)*. IEEE. 2018, pp. 197–202 (cited on pages 19, 23, 24, 29, 81).

[60] I Alaoui Ismaili et al. 'Comparative Study of ZigBee and 6LoWPAN Protocols'. In: *ICCWCS 2019: Third International Conference on Computing and Wireless Communication Systems, ICCWCS 2019, April 24–25, 2019, Faculty of Sciences, Ibn Tofail University-Kénitra-Morocco*. European Alliance for Innovation. 2019, p. 264 (cited on pages 19, 21).

[61] Fan Wu, Taiyang Wu, and Mehmet Rasit Yuce. 'An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications'. In: *Sensors* 19.1 (2018), p. 21 (cited on page 19).

[62] Market Intelligence & Consulting Institute (MIC). 'Global Wi-Fi Enabled Devices Shipment Forecast, 2020-2024'. In: *Market Intelligence & Consulting Institute (MIC)* (2020) (cited on page 20).

[63] Jekishan K Parmar and Ankit Desai. 'IoT: Networking Technologies and Research Challenges'. In: *International Journal of Computer Applications* 154.7 (2016), pp. 1–6 (cited on pages 20, 73).

[64] Umber Noreen, Ahcène Bounceur, and Laurent Clavier. 'A Study of LoRa Low Power and Wide Area Network Technology'. In: *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE. 2017, pp. 1–6 (cited on pages 20, 21, 96).

[65] Evgeny Khorov et al. 'Enabling the Internet of Things with Wi-Fi Halow—Performance Evaluation of the Restricted Access Window'. In: *IEEE Access* 7 (2019), pp. 127402–127415 (cited on pages 20, 21).

[66] Matthew Gast. *802.11 Wireless Networks: The Definitive Guide*. " O'Reilly Media, Inc.", 2005 (cited on page 21).

[67] Carles Gomez, Joaquim Oller, and Josep Paradells. 'Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-power Wireless Technology'. In: *sensors* 12.9 (2012), pp. 11734–11753 (cited on page 21).

[68] IEEE Computer Society. 'Part 15 . 4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)'. In: *IEEE Standard* (2006), pp. 1–26 (cited on page 21).

[69] Geoff Mulligan. 'The 6LoWPAN Architecture'. In: *Proceedings of the 4th workshop on Embedded networked sensors*. 2007, pp. 78–82 (cited on page 21).

[70] Xavier Vilajosana et al. 'IETF 6TiSCH: A Tutorial'. In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 595–615 (cited on page 21).

[71] Ermal Elbasani, Pattamaset Siriporn, and Jae Sung Choi. 'A Survey on RFID in Industry 4.0'. In: *Internet of Things for Industry 4.0: Design, Challenges and Solutions* (2020), pp. 1–16 (cited on page 22).

[72] Vijay Pillai et al. 'An Ultra-low-power Long Range Battery/passive RFID Tag for UHF and Microwave Bands with a Current Consumption of 700 nA at 1.5 V'. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 54.7 (2007), pp. 1500–1512 (cited on page 22).

[73] Brandon Foubert and Nathalie Mitton. 'Long-range Wireless Radio Technologies: A Survey'. In: *Future internet* 12.1 (2020), p. 13 (cited on pages 22, 24, 26).

[74] Albert Berni and WO Gregg. 'On the Utility of Chirp Modulation for Digital Signaling'. In: *IEEE Transactions on Communications* 21.6 (1973), pp. 748–751 (cited on page 22).

[75] Ferran Adelantado et al. 'Understanding the Limits of LoRaWAN'. In: *IEEE Communications magazine* 55.9 (2017), pp. 34–40 (cited on pages 23, 73).

[76] J Schlienz and D Raddino. 'Narrowband Internet of Things Whitepaper'. In: *White Paper, Rohde&Schwarz* (2016), pp. 1–42 (cited on page 24).

[77] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. 'A Survey on LPWA Technology: LoRa and NB-IoT'. In: *Ict Express* 3.1 (2017), pp. 14–21 (cited on pages 24, 29, 81, 96).

[78] Jeffrey G Andrews et al. 'What will 5G be?' In: *IEEE Journal on selected areas in communications* 32.6 (2014), pp. 1065–1082 (cited on page 25).

[79] Rupendra Nath Mitra and Dharma P Agrawal. '5G Mobile Technology: A Survey'. In: *ICT express* 1.3 (2015), pp. 132–137 (cited on page 25).

[80] Mostafa Zaman Chowdhury et al. '6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions'. In: *IEEE Open Journal of the Communications Society* 1 (2020), pp. 957–975 (cited on page 25).

[81] Dinh C Nguyen et al. '6G Internet of Things: A Comprehensive Survey'. In: *IEEE Internet of Things Journal* 9.1 (2021), pp. 359–383 (cited on page 25).

[82] Kah Yung Yap, Hon Huin Chin, and Jiri Jaromir Klemes. 'Future Outlook on 6G Technology for Renewable Energy Sources (RES)'. In: *Renewable and Sustainable Energy Reviews* 167 (2022), p. 112722 (cited on page 25).

[83] Anutusha Dogra, Rakesh Kumar Jha, and Shubha Jain. 'A Survey on Beyond 5G Network with the Advent of 6G: Architecture and Emerging Technologies'. In: *IEEE Access* 9 (2020), pp. 67512–67547 (cited on page 25).

[84] Fadi M Al-Turjman, Muhammad Imran, and Sheikh Tahir Bakhsh. 'Energy Efficiency Perspectives of Femtocells in Internet of Things: Recent Advances and Challenges'. In: *IEEE Access* 5 (2017), pp. 26808–26818 (cited on page 26).

[85] Li Zhen et al. 'Energy-efficient Random Access for LEO Satellite-assisted 6G Internet of Remote Things'. In: *IEEE Internet of Things Journal* 8.7 (2020), pp. 5114–5128 (cited on page 26).

[86] Amrit Mukherjee et al. 'Energy-efficient Resource Allocation Strategy in Massive IoT for Industrial 6G Applications'. In: *IEEE Internet of Things Journal* 8.7 (2020), pp. 5194–5201 (cited on page 26).

[87] Ali Hassan Sodhro et al. 'Toward 6G Architecture for Energy-efficient Communication in IoT-enabled Smart Automation Systems'. In: *IEEE Internet of Things Journal* 8.7 (2020), pp. 5141–5148 (cited on page 26).

[88] Nan Zhao et al. 'Exploiting Interference for Energy Harvesting: A Survey, Research Issues, and Challenges'. In: *IEEE Access* 5 (2017), pp. 10403–10421 (cited on page 26).

[89] URL: https://www.ingenu.com/technology/rpma/competition/ (cited on page 26).

[90] URL: https://www.openweightless.org/ (cited on page 26).

[91] URL: https://www.dash7-alliance.org/ (cited on page 26).

[92] Den Shay. 'DECT 2020 NR : Sustainability and Scale for the IoT (White Paper)'. In: *ABI Research, Wirepas* (2020) (cited on page 27).

[93] Sam Lucero. 'Satellite IoT Market Report 2020'. In: *Omdia (White Paper)* (2020) (cited on page 27).

[94] Marco Centenaro et al. 'A Survey on Technologies, Standards and Open Challenges in Satellite IoT'. In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), pp. 1693–1720 (cited on page 27).

[95] Abdul Ahad et al. 'Technologies Trend towards 5G Network for Smart Health-Care using IoT: A Review'. In: *Sensors* 20.14 (2020), p. 4047 (cited on pages 27, 73).

[96] Peter Kaňuch, Dominik Macko, and Ladislav Hudec. 'Survey: Classification of the IoT Technologies for Better Selection to Real Use'. In: *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2020, pp. 500–505 (cited on pages 29, 81).

[97] Augustine Ikpehai et al. 'Low-power Wide Area Network Technologies for Internet-of-Things: A Comparative Review'. In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 2225–2240 (cited on pages 29, 81).

[98] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer. 'A Comparative Study of LPWAN Technologies for Large-scale IoT Deployment'. In: *ICT express* 5.1 (2019), pp. 1–7 (cited on pages 29, 73, 81).

[99] Cedric Adjih et al. 'FIT IoT-LAB: A Large Scale Open Eperimental IoT Testbed'. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE. 2015, pp. 459–464 (cited on pages 30, 31, 99).

[100] Luis Sanchez et al. 'SmartSantander: IoT Experimentation over a Smart City Testbed'. In: *Computer Networks* 61 (2014), pp. 217–238 (cited on pages 30, 31).

[101] Stefan Bouckaert et al. 'The w-iLab. t Testbed'. In: *Testbeds and Research Infrastructures. Development of Networks and Communities: 6th International ICST Conference, TridentCom 2010, Berlin, Germany, May 18-20, 2010, Revised Selected Papers 6*. Springer. 2011, pp. 145–154 (cited on pages 30, 31).

[102] Maxim Chernyshev et al. 'Internet of Things (IoT): Research, Simulators, and Testbeds'. In: *IEEE Internet of Things Journal* 5.3 (2017), pp. 1637–1647 (cited on page 31).

[103] Yuuichi Teranishi et al. 'JOSE: An Open Testbed for Field Trials of Large-scale IoT Services'. In: *Journal of the National Institute of Information and Communications Technology* 62.2 (2016), pp. 151–159 (cited on page 31).

[104] Ioannis Chatzigiannakis et al. 'WISEBED: An Open Large-scale Wireless Sensor Network Testbed'. In: *Sensor Applications, Experimentation, and Logistics: First International Conference, SENSAPPEAL 2009, Athens, Greece, September 25, 2009, Revised Selected Papers 1*. Springer. 2010, pp. 68–87 (cited on page 31).

[105] Jonathan Munoz et al. 'OpenTestBed: Poor Man's IoT Testbed'. In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2019, pp. 467–471 (cited on page 31).

[106] András Varga. 'Discrete Event Simulation System'. In: *Proc. of the European Simulation Multiconference (ESM'2001)*. 2001, pp. 1–7 (cited on pages 32–34).

[107] Vinko Erceg et al. 'An Empirically based Path Loss Model for Wireless Channels in Suburban Environments'. In: *IEEE Journal on selected areas in communications* 17.7 (1999), pp. 1205–1211 (cited on page 32).

[108] Harald T Friis. 'A Note on a Simple Transmission Formula'. In: *Proceedings of the IRE* 34.5 (1946), pp. 254–256 (cited on pages 32, 33).

[109] Eraldo Damosso. *Digital Mobile Radio towards Future Generation Systems: COST action 231*. European Commission, 1999 (cited on page 33).

[110] Isabona Joseph and CC Konyeha. 'Urban Area Path Loss Propagation Prediction and Optimisation using HATA Model at 800MHz'. In: *IOSR Journal of Applied Physics (IOSR-JAP)* 3.4 (2013), pp. 8–18 (cited on page 33).

[111] Theodore S Rappaport et al. 'Wireless Communications: Past Events and a Future Perspective'. In: *IEEE communications Magazine* 40.5 (2002), pp. 148–161 (cited on page 33).

[112] Borja Martinez et al. 'The Power of Models: Modeling Power Consumption for IoT Devices'. In: *IEEE Sensors Journal* 15.10 (2015), pp. 5777–5789 (cited on pages 33, 38, 41).

[113] Thomas R Henderson et al. 'Network Simulations with the ns-3 Simulator'. In: *SIGCOMM demonstration* 14.14 (2008), p. 527 (cited on pages 33, 34).

[114] Kamal Mehdi et al. 'Cupcarbon: A Multi-Agent and Discrete Event Wireless Sensor Network Design and Simulation Tool'. In: *7th International ICST Conference on Simulation Tools and Techniques, Lisbon, Portugal, 17-19 March 2014*. Institute for Computer Science, Social Informatics and Telecommunications. 2014, pp. 126–131 (cited on pages 33, 34).

[115] Sameera Abar et al. 'Agent Based Modelling and Simulation tools: A Review of the State-of-art Software'. In: *Computer Science Review* 24 (2017), pp. 13–33 (cited on page 34).

[116] Fredrik Osterlind et al. 'Cross-level Sensor Network Simulation with Cooja'. In: *Proceedings. 2006 31st IEEE conference on local computer networks*. IEEE. 2006, pp. 641–648 (cited on page 34).

[117] Sorout Dinesh and Goyal Sonal. 'Qualnet Simulator'. In: *International Journal of Information & Computation Technology, ISSN* (2014), pp. 0974–2239 (cited on page 34).

[118] Starting Matlab. 'Matlab'. In: *The MathWorks, Natick, MA* (2012) (cited on page 34).

[119] Martin Heusse et al. 'Performance Anomaly of 802.11 b'. In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*. Vol. 2. IEEE. 2003, pp. 836–843 (cited on pages 35, 41).

[120] Yuyi Sun et al. 'Throughput Modeling and Analysis of Random Access in Narrowband Internet of Things'. In: *IEEE Internet of Things Journal* 5.3 (2017), pp. 1485–1493 (cited on pages 35, 41).

[121] LoRa Alliance Technical Commitee. 'LoRaWAN 1.1 Specification'. In: *Technical Document* (2017), pp. 1–101 (cited on pages 36, 41).

[122] Asif M Yousuf et al. 'Throughput, Coverage and Scalability of LoRa LPWAN for Internet of Things'. In: *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE. 2018, pp. 1–10 (cited on pages 36, 41).

[123] Morteza Mehrnoush et al. 'Analytical Modeling of Wi-Fi and LTE-LAA Coexistence: Throughput and Impact of Energy Detection Threshold'. In: *IEEE/ACM Transactions on Networking* 26.4 (2018), pp. 1990–2003 (cited on pages 36, 41).

[124] René Brandborg Sørensen et al. 'Analysis of Latency and MAC-layer Performance for Class A LoRaWAN'. In: *IEEE Wireless Communications Letters* 6.5 (2017), pp. 566–569 (cited on pages 37, 41).

[125] Semtech Corporation. 'SX1272/3/6/7/8 LoRa Modem Design Guide, AN1200.13'. In: July (2013), p. 9 (cited on pages 37, 41).

[126] Mohieddine El Soussi et al. 'Evaluating the Performance of eMTC and NB-IoT for Smart City Applications'. In: *2018 ieee international conference on communications (icc)*. IEEE. 2018, pp. 1–7 (cited on pages 37, 41).

[127] Mare Srbinovska, Vladimir Dimcev, and Cvetan Gavrovski. 'Energy Consumption Estimation of Wireless Sensor Networks in Greenhouse Crop Production'. In: *IEEE EUROCON 2017-17th international conference on smart technologies*. IEEE. 2017, pp. 870–875 (cited on pages 39, 41).

[128] Lluis Casals et al. 'Modeling the Energy Performance of LoRaWAN'. In: *Sensors* 17.10 (2017), p. 2364 (cited on pages 39, 41, 52, 96).

[129] Pilar Andres-Maldonado et al. 'Analytical Modeling and Experimental Validation of NB-IoT Device Energy Consumption'. In: *IEEE Internet of Things Journal* 6.3 (2019), pp. 5691–5701 (cited on pages 39–41).

[130] Pablo Serrano et al. 'Per-frame Energy Consumption in 802.11 Devices and its Implication on Modeling and Design'. In: *IEEE/ACM Transactions on networking* 23.4 (2014), pp. 1243–1256 (cited on pages 40, 41, 55, 95).

[131] Yandja Lalle et al. 'A Comparative Study of Lorawan, Sigfox, and NB-IoT for Smart Water Grid'. In: *2019 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE. 2019, pp. 1–6 (cited on pages 41, 81).

[132] Richard Verhoeven, Stash Kempinski, and Nirvana Meratnia. 'Performance Evaluation of Wi-Fi Halow, NB-IoT and LoRa for Smart City Applications'. In: *Proceedings of the 19th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. 2022, pp. 17–24 (cited on pages 41, 81).

[133] Benny Vejlgaard et al. 'Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT'. In: *2017 IEEE 85th vehicular technology conference (VTC Spring)*. IEEE. 2017, pp. 1–5 (cited on pages 42, 81).

[134] Farooq Bari and Victor Leung. 'Multi-Attribute Network Selection by Iterative TOPSIS for Heterogeneous Wireless Access'. In: *2007 4th IEEE consumer communications and networking conference*. IEEE. 2007, pp. 808–812 (cited on pages 42, 81).

[135] Farooq Bari and Victor CM Leung. 'Automated Network Selection in a Heterogeneous Wireless Network Environment'. In: *IEEE network* 21.1 (2007), pp. 34–40 (cited on pages 42, 81).

[136] Mohamed Abdelkrim Senouci et al. 'TOPSIS-based Dynamic Approach for Mobile Network Interface Selection'. In: *Computer Networks* 107 (2016), pp. 304–314 (cited on pages 42, 81).

[137] Gwo-Hshiung Tzeng and Jih-Jeng Huang. *Multiple Attribute Decision Making: Methods and Applications*. CRC press, 2011 (cited on pages 42, 68).

[138] Gazis, Vangelis, et al. 'Toward a Generic" Always Best Connected" Capability in Integrated WLAN/UMTS Cellular Mobile Networks (and Beyond)'. In: *IEEE Wireless Communications* (2005) (cited on page 42).

[139] Lusheng Wang and David Binet. 'Best Permutation: A Novel Network Selection Scheme in Heterogeneous Wireless Networks'. In: *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*. 2009 (cited on page 42).

[140] Jindong Hou and Dominic C O'Brien. 'Vertical Handover-Decision-Making Algorithm using Fuzzy Logic for the Integrated Radio-and-OW System'. In: *IEEE Transactions on Wireless Communications* (2006) (cited on page 42).

[141] Wenhui Zhang. 'Handover Decision using Fuzzy MADM in Heterogeneous Networks'. In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2004 (cited on page 42).

[142] Stevens-Navarro, Enrique, et al. 'An MDP-based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks'. In: *IEEE Transactions on Vehicular Technology* (2008) (cited on page 42).

[143] Ning, Zhaolong, et al. 'Markov-based Vertical Handoff Decision Algorithms in Heterogeneous Wireless Networks'. In: *Computers & Electrical Engineering* (2014) (cited on page 42).

[144] Matteo Cesana, Nicola Gatti, and Ilaria Malanchini. 'Game Theoretic Analysis of Wireless Access Network Selection: Models, Inefficiency Bounds, and Algorithms'. In: *Proceedings of the 3rd international conference on performance evaluation methodologies and tools*. 2008 (cited on page 42).

[145] Liu, Bin, et al. 'AHP and Game Theory based Approach for Network Selection in Heterogeneous Wireless Networks'. In: *IEEE 11th Consumer Communications and Networking Conference (CCNC)*. 2014 (cited on page 42).

[146] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. 'Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations'. In: *IEEE/ACM Transactions on Networking* (2012) (cited on page 42).

[147] Henri, Sébastien, et al. 'Multi-Armed Bandit in Action: Optimizing Performance in Dynamic Hybrid Networks'. In: *IEEE/ACM Transactions on Networking* (2018) (cited on page 42).

[148] Bilal R Al-Kaseem et al. 'A New Intelligent Approach for Optimizing 6LoWPAN MAC Layer Parameters'. In: *IEEE Access* 5 (2017), pp. 16229–16240 (cited on page 43).

[149] Moussa Aboubakar et al. 'An Efficient and Adaptive Configuration of IEEE 802.15. 4 MAC for Communication Delay Optimisation'. In: *2020 11th International Conference on Network of the Future (NoF)*. IEEE. 2020, pp. 1–7 (cited on page 43).

[150] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. 'An Online Learning Approach for Auto Link-Configuration in IEEE 802.11 AC Wireless Networks'. In: *Computer Networks* 181 (2020), p. 107426 (cited on page 43).

[151] Chen Chen et al. 'Contention Resolution in Wi-Fi 6-enabled Internet of Things based on Deep Learning'. In: *IEEE Internet of Things Journal* 8.7 (2020), pp. 5309–5320 (cited on page 43).

[152] Daniel Camps-Mur et al. 'Leveraging 802.11 n Frame Aggregation to Enhance QoS and Power Consumption in Wi-Fi Networks'. In: *Computer Networks* 56.12 (2012), pp. 2896–2911 (cited on page 43).

[153] Nick Rozanski and Eoin Woods. *Software Systems Architecture: Working with Stakeholders using Viewpoints and Perspectives*. Addison-Wesley, 2011 (cited on page 46).

[154] Mirko Stoffers and George Riley. 'Comparing the ns-3 Propagation Models'. In: *2012 IEEE 20th international symposium on modeling, analysis and simulation of computer and telecommunication systems*. IEEE. 2012, pp. 61–67 (cited on pages 47, 48).

[155] Davide Magrin, Marco Centenaro, and Lorenzo Vangelista. 'Performance Evaluation of LoRa Networks in a Smart City Scenario'. In: *2017 IEEE International Conference on communications (ICC)*. ieee. 2017, pp. 1–7 (cited on pages 52, 53, 73, 96).

[156] He Wu, Sidharth Nabar, and Radha Poovendran. 'An Energy Framework for the Network Simulator 3 (ns-3)'. In: *4th International ICST Conference on Simulation Tools and Techniques*. 2012 (cited on pages 55, 95).

[157] Albert Rego et al. 'An Intelligent System for Video Surveillance in IoT Environments'. In: *IEEE Access* 6 (2018), pp. 31580–31598 (cited on page 57).

[158] Video trace library. http://trace.eas.asu.edu/. Arizona State University (cited on page 57).

[159] Nedis. 'SmartLife Outdoor Camera'. In: () (cited on page 58).

[160] Joseph Benin, Michael Nowatkowski, and Henry Owen. 'Vehicular Network Simulation Propagation Loss Model Parameter Standardization in ns-3 and Beyond'. In: *2012 Proceedings of IEEE Southeastcon*. IEEE. 2012, pp. 1–5 (cited on page 60).

[161] Wi-Fi code. https://github.com/SamirSim/wifi-icc (cited on page 60).

[162] LoRa code. https://github.com/SamirSim/lora-icc (cited on page 60).

[163] Beecham Research. 'Why IoT projects fail & how to beat the odds An executive summary'. In: (2020) (cited on page 61).

[164] Microsoft. 'IoT Signals - Manufacturing Spotlight'. In: (2022) (cited on page 61).

[165] A Aramayo-Prudencio et al. 'Digital Manufacturing–Escaping Pilot Purgatory'. In: *Digital McKinsey, digital-manufacturing-escaping-pilot-purgatory. pdf (mckinsey. com)* (2018) (cited on pages 61, 106).

[166] Tim Winter et al. *RPL: IPv6 Routing Protocol for Low-power and Lossy Networks*. Tech. rep. 2012 (cited on page 62).

[167] Aditya Chauhan and Rahul Vaish. 'A Comparative Study on Decision Making Methods with Interval Data'. In: *Journal of Computational Engineering* 2014 (2014) (cited on page 71).

[168] HINTS source code. https://github.com/SamirSim/Selection-Methodology-IoT (cited on pages 72, 82).

[169] IEEE Computer Society LAN/MAN Standards Committee et al. 'IEEE Standard for Information technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications'. In: *IEEE Std 802.11^* (2007) (cited on page 73).

[170] Weiping Sun, Munhwan Choi, and Sunghyun Choi. 'IEEE 802.11ah: A Long Range 802.11 WLAN at Sub 1 GHz'. In: *J. ICT Stand.* 1 (2013), pp. 83–108 (cited on page 73).

[171] Le Tian et al. 'Implementation and Validation of an IEEE 802.11 ah Module for ns-3'. In: *Proceedings of the Workshop on ns-3*. 2016, pp. 49–56 (cited on page 73).

[172] Marco Mezzavilla et al. 'End-to-end Simulation of 5G mmWave Networks'. In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2237–2263 (cited on page 73).

[173] Victor Baños-Gonzalez et al. 'IEEE 802.11 ah: A Technology to Face the IoT Challenge'. In: *Sensors* 16.11 (2016), p. 1960 (cited on page 73).

[174] Kaveh Pahlavan and Prashant Krishnamurthy. 'Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective'. In: *International Journal of Wireless Information Networks* 28.1 (2021), pp. 3–19 (cited on page 73).

[175] Bahzad Taha Jijo et al. 'A Comprehensive Survey of 5G mm-wave Technology Design Challenges'. In: *Asian Journal of Research in Computer Science* 8.1 (2021), pp. 1–20 (cited on page 74).

[176] Mohammad H Mazaheri et al. 'A millimeter wave Network for Billions of Things'. In: *Proceedings of the ACM Special Interest Group on Data Communication*. 2019, pp. 174–186 (cited on page 77).

[177] Brandon Foubert and Nathalie Mitton. 'RODENT: A Flexible TOPSIS based Routing Protocol for Multi-technology Devices in Wireless Sensor Networks'. In: *ITU Journal on Future and Evolving Technologies* 2.1 (2021) (cited on page 81).

[178] Krzysztof Pawlikowski, H-DJ Jeong, and J-SR Lee. 'On Credibility of Simulation Studies of Telecommunication Networks'. In: *IEEE Communications magazine* 40.1 (2002), pp. 132–139 (cited on page 85).

[179] Ching-Lai Hwang and Kwangsun Yoon. 'Methods for Multiple Attribute Decision Making'. In: *Multiple attribute decision making*. Springer, 1981, pp. 58–191 (cited on page 87).

[180] George AF Seber and Alan J Lee. *Linear Regression Analysis*. Vol. 330. John Wiley & Sons, 2003 (cited on page 89).

[181] Jerome H Friedman. 'Greedy Function Approximation: A Gradient Boosting Machine'. In: *Annals of statistics* (2001), pp. 1189–1232 (cited on page 89).

[182] Leo Breiman. 'Random Forests'. In: *Machine learning* 45 (2001), pp. 5–32 (cited on page 89).

[183] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 'Extremely Randomized Trees'. In: *Machine learning* 63 (2006), pp. 3–42 (cited on page 89).

[184] Leif E Peterson. 'K-Nearest Neighbor'. In: *Scholarpedia* 4.2 (2009), p. 1883 (cited on page 89).

[185] Alex J Smola and Bernhard Schölkopf. 'A Tutorial on Support Vector Regression'. In: *Statistics and computing* 14 (2004), pp. 199–222 (cited on page 89).

[186] Argha Sen et al. 'An ns3-based Energy Module of 5G NR User Equipments for Millimeter Wave Networks'. In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2021, pp. 1–2 (cited on page 96).

[187] Carles Gomez et al. 'A Sigfox Energy Consumption Model'. In: *Sensors* 19.3 (2019), p. 681 (cited on page 96).

[188] Philipp Hurni and Torsten Braun. 'Calibrating Wireless Sensor Network Simulation Models with Real-world Experiments'. In: *NETWORKING 2009: 8th International IFIP-TC 6 Networking Conference, Aachen, Germany, May 11-15, 2009. Proceedings 8*. Springer. 2009, pp. 1–13 (cited on page 100).

[189] Feng Chen et al. 'An Energy Model for Simulation Studies of Wireless Sensor Networks using OMNeT++'. In: (2009) (cited on page 101).

[190] Andriy Stetsko, Martin Stehlik, and Vashek Matyas. 'Calibrating and Comparing Simulators for Wireless Sensors Networks'. In: *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*. IEEE. 2011, pp. 733–738 (cited on page 101).

[191] William Koch. 'An Approach for Automating the Calibration of Simulations of Parallel and Distributed Computing Systems'. PhD thesis. University of Hawai'i at Manoa, 2021 (cited on page 101).

[192] Peter Öhlén et al. 'Network Digital Twins–Outlook and Opportunities'. In: *Ericsson Technology Review* 2022.12 (2022), pp. 2–11 (cited on page 103).

[193] Daler N Rakhmatov and Sarma BK Vrudhula. 'An Analytical High-level Battery Model for Use in Energy Management of Portable Electronic Systems'. In: *IEEE/ACM International Conference on Computer Aided Design. ICCAD 2001. IEEE/ACM Digest of Technical Papers (Cat. No. 01CH37281)*. IEEE. 2001, pp. 488–493 (cited on page 110).

[194] Muhammad Naeem et al. 'A Sigfox Module for the Network Simulator 3'. In: *Proceedings of the 2022 Workshop on ns-3*. 2022, pp. 81–88 (cited on page 119).

[195] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. 'A Flexible Framework for Multi-objective Bayesian Optimization using Random Scalarizations'. In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 766–776 (cited on page 120).

[196] David A Van Veldhuizen, Gary B Lamont, et al. 'Evolutionary Computation and Convergence to a Pareto Front'. In: *Late breaking papers at the genetic programming 1998 conference*. Citeseer. 1998, pp. 221–228 (cited on page 120).

Special Terms

Numbers
3GPP 3rd Generation Partnership Project. 23, 24
6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks. 21, 49, 72, 73, 91, 92
6TiSCH IPv6 over the Time Slotted Channel Hopping. 21

A
ACK Acknowledge. 50
ARPANET Advanced Research Project Agency Network. 14

B
BLE Bluetooth Low Energy. 21, 31
BMS Building Management System. 2, 104
BPSK Binary Phase Shift Keying. 23

C
CDMA Code Division Multiple Access. 26
CR Coding Rate. 49, 50, 79, 80, 91, 92
CRC Cyclic Redundancy Check. 49, 50, 91, 92
CSMA Carrier Sense Multiple Access . 20, 21, 49, 50
CSS Chirp Spread Spectrum. 22

D
D7AP DASH7 Alliance Protocol. 26
DARPA Defense Advanced Research Projects Agency. 14
DSSS Direct Sequence Spread Spectrum. 21

E
ED End-device. 73
ETSI European Telecommunications Standards Institute. 26

F
FLIP Frugal Labs IoT Platform. 16

G
GFSK Gaussian Frequency Shift Keying. 21
GMSK Gaussian Minimum Shift Keying. 24, 26
GUI Graphical User Interface. 33
GW Gateway. 73, 75, 78, 80

H
HARQ Hybrid Automatic Request. 50, 76
HF High Frequency. 22
HVAC Heating, ventilation and air conditiong. 2, 15, 104

I
IaaS Infrastructure as a Service. 31
IoT Internet of Things. iii, 1, 13
ISM Industrial, Scientific and Medical. 21, 22, 26
ITU International Telecommunications Union. 1

J

JOSE Japan-wide Orchestrated Smart/Sensor Environment. 31

K

KPI Key Performance Indicators. v, ix, x, 35, 36, 40–46, 51, 53, 59–70, 72–79, 81–83, 85–90, 92–95, 101, 105–110, 112, 121

L

LF Low Frequency. 22
LPWAN Low Power Wide Area Network. 15, 22, 26, 29, 119
LR-WPAN Low Rate Wireless Personal Area Network. 21
LTE Long Term Evolution. 24, 31
LTE-LAA LTE-Licensed Assisted Access. 36

M

MAC Medium Access Layer. 22, 28, 34, 35, 43, 119
MADM Multi Attibute Decision Making. 9, 61, 68, 81, 82, 87, 117, 120
MCS Modulation and Coding Scheme. 20, 40, 43, 48–50, 55, 57, 58, 60, 63, 88, 92, 93
MDP Markov Decision Process. 42
MIMO Multiple Input Multiple Output. 43
ML Machine Learning. 15, 42, 85
mMTC Massive Machine Type Communications. 26
mmWave Millimeter Wave. 25
MTD Machine Type Devices. 27
MW Microwave Frequency. 22

N

NB-IoT Narrowband-IoT. ix, 24, 35, 37, 39, 40
NDT Network Digital Twin. 103, 121
NIC Network Interface Card. ix, 40, 53, 95, 99
ns-3 Network Simulator 3. ix, x, 33, 41, 51–53, 55–57, 73, 81, 86, 87, 95–97, 99, 100, 103, 105, 106, 108, 110, 115, 119

O

OFDM Orthogonal Frequency-Division Multiplexing. 25, 26
OFDMA Orthogonal Frequency-Division Multiple Access. 24, 25
OS Operating System. 30, 34, 40
OSI Open Systems Interconnection. 19

P

P2P Point to Point. 20, 21
pcap Packet Capture. 110
PDC Parallel and Distributed Computing. 101
PHY Physical. 28, 119
PoC Proof of Concept. iii, 29, 85, 95–97, 103, 104, 118, 120, 121

Q

QoS Quality of Service. 2, 6, 43, 103
QPSK Quadrature Phase Shift Keying. 24, 26

R

RA Random Access. 35
RAW Random Access Window. 49
RF Radio Frequency. 21, 39
RFID Radio-Frequency Identification. 14, 15, 22, 30
RLC-AM Radio Ling Control Acknowledged Mode. 50, 76
ROI Return on Investment. 3

RPL Routing Protocol for Low-Power and Lossy Networks. 21, 62, 119
RPMA Random Phase Multiple Access. 26
RTOS Real-Time Operating System. 18

S
SAW Simple Additive Weighting. 42
SC-FDMA Single-Carrier Frequency-Division Multiple Access. 24
SF Spreading Factor. 22, 36–38, 48, 49, 53, 54, 59, 63, 65, 66, 79, 80, 85, 87, 88, 91, 92, 112

T
TDMA Time Division Multiple Access. 21
TOPSIS Technique for Order Preference Similarity to Ideal Solution. ix, 42, 68, 71, 82, 89, 117, 120

U
UHF Ultra High Frequency. 22
UNB Ultra Narrow Band. 26

V
VHO Vertical Handoff. 42

W
Weightless-SIG Weightless Special Interest Group. 26
WPAN Wireless Personal Area Network. 20
WPM Weighted Product Methods. 42
WSN Wireless Sensor Network. 15, 16, 33, 34, 39, 52, 100, 101