Universitat Politècnica de Catalunya

Doctoral Thesis

Ambient Intelligence in buildings. Design and development of an interoperable Internet of Things platform

Author:
David Sembroiz Ausejo

Supervisors:
Dr. Davide Careglio
Dr. Sergio Ricciardi

A thesis submitted in fulfillment of the requirements
for the PhD programme in Computer Architecture and Technology

in the

Broadband Communications Research Group
Computer Architecture Department

January 16, 2020

Declaration of Authorship

I, David SEMBROIZ AUSEJO, declare that this thesis titled, “Ambient Intelligence in buildings. Design and development of an interoperable Internet of Things platform” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:
____

Date:
____

Abstract

David Sembroiz Ausejo

Ambient Intelligence in buildings. Design and development of an interoperable Internet of Things platform

During many years, people and governments have been warned about the increasing levels of pollution and greenhouse gases (GHG) emissions that are endangering our lives on this planet. The Information and Communication Technology sector, usually known as the ICT sector, responsible for the computerisation of the society, has been pinpointed as one of the most important sectors contributing to such problem. Many efforts, however, have been put to shift the trend towards the utilisation of renewable resources, such as wind or solar power. Even though governments have agreed to follow this path and avoid the usage of non-renewable energies, it is not enough.

Although the ICT sector might seem an added problem due to the number of connected devices, technology improvements and hardware optimisation enable new ways of fighting against global warming and GHG emissions.

During the last decades, public and private companies aware of the current problem have focused their efforts on creating energy-aware and energy-efficient solutions both in terms of hardware and software. New Internet physical backbone networks are being created with energy-efficient devices and ISPs tend to create energy-aware systems inside their domain.

Following this tendency, any new system created and added to the network must guarantee a certain level of energy-awareness and optimal resource management.

The aforementioned computerisation has forced companies to evolve their work into a computer-assisted one. Due to this, companies are now forced to establish their main headquarters inside buildings for work coordination, connection and management. Due to this, buildings are becoming one of the most important issues regarding energy consumption.

In order to cope with such problem, the Internet of Things (IoT) offers new paradigms and alternatives for leading the change. IoT is commonly defined as the network of physical and virtual objects that are capable of collecting surrounding data and exchanging it between them or through the Internet. Thanks to these networks, it is possible to monitor any thinkable metric inside buildings, and, then, utilise this information to build efficient automated systems, commonly known as Building Energy Management Systems (BEMS), capable of extracting conclusions on how to optimally and efficiently manage the resources of the building. ICT companies have foreseen this market opportunity that, paired with the appearance of smaller, efficient and more durable sensors, allows the development of efficient IoT systems. However, the lack of agreement and standardisation creates chaos inside IoT, and the horizontal connectivity between such systems is still a challenge. Moreover, the vast amount of data to process requires the utilisation of Big Data techniques to guarantee close to real-time responses.

This thesis initially presents a standard Cloud-based IoT architecture that tries to cope with the aforementioned problems by employing a Cloud middleware that obfuscates the underlying hardware architecture and permits the aggregation of data from multiple heterogeneous sources. Also, sensor information is exposed to any third-party client after authentication.

The utilisation of automated IoT systems for managing building resources requires high reliability, resilience, and availability. The loss of sensor data is not permitted due to the negative consequences it might have, such as disruptive resource management. For this, it is mandatory to grant backup options to sensor networks in order to guarantee correct functioning in case of partial network disconnections. Additionally, the placement of the sensors inside the building must guarantee minimal energy consumption while fulfilling sensing requirements. For this, we enhance the placement problem of heterogeneous Wireless Sensor Networks by adding clustering constraints to ensure that every sensor type reaches its corresponding metric, and protection by enabling multiple transmission paths for data delivery. Resilience is also studied in large multi-hop homogeneous WSN by identifying the most critical sensors.

Finally, a building resource management use case is presented by means of a simulation tool. The tool draws on occupants' probabilistic models and environmental condition models for actuating upon building elements to ensure optimal and efficient functioning. Occupants' comfort is also taken into consideration and the trade-off between the two metrics is studied.

All the presented work is meant to deliver insights and tools for current and future IoT system implementations by setting the basis for standardisation agreements yet to happen.

Resumen

David Sembroiz Ausejo

Ambient Intelligence in buildings. Design and development of an interoperable Internet of Things platform

Durante muchos años, se ha alertado a la población y a los gobiernos acerca del incremento en los niveles de polución y de emisión de gases de efecto invernadero, que están poniendo en peligro nuestra vida en la Tierra. El sector de las Tecnologías de la Información y Comunicación, normalmente conocido como las TIC, responsable de la informatización de la sociedad, ha sido señalada como uno de los sectores más importantes encargado de agravar tal problema. Sin embargo, mucho esfuerzo se está poniendo para revertir esta situación mediante el uso de recursos renovables, como la energía eólica o solar. A pesar de que los gobiernos han acordado seguir dicho camino y evitar el uso de energía no renovable tanto como sea posible, no es suficiente para erradicar el problema.

Aunque el sector de las TIC pueda parecer un problema añadido dada la gran cantidad y el incremento de dispositivos conectados, las mejoras en tecnología y en hardware están habilitando nuevas maneras de luchar contra el calentamiento global y la emisión de gases de efecto invernadero.

Durante las últimas décadas, compañías del sector público y privado conscientes del problema han centrado sus esfuerzos en la creación de soluciones orientadas a la eficiencia energética tanto a nivel de hardware como de software. Las nuevas redes troncales están siendo creadas con dispositivos eficientes y los proveedores de servicios de Internet tienden a crear sistemas conscientes de la energía para su optimización dentro de su dominio.

Siguiendo esta tendencia, cualquier nuevo sistema creado y añadido a la red debe garantizar un cierto nivel de conciencia y un manejo óptimo de los recursos que utiliza.

La informatización, anteriormente mencionada, ha forzado a las empresas a evolucionar su modelo de negocio hacia uno más enfocado en la utilización de redes de ordenadores para gestionar sus recursos. Por eso, dichas compañías se están viendo forzadas a establecer sus sedes centrales dentro de edificios, para tener un mayor control sobre la coordinación, conexión y manejo de sus recursos. Esto está provocando un aumento en el consumo energético de los edificios, que se están convirtiendo en uno de los principales problemas.

Para poder hacer frente al problema, el Internet de las Cosas o Internet of Things (IoT) ofrece nuevos paradigmas y alternativas para liderar el cambio. IoT se define como la red de objetos físicos y virtuales, capaces de recolectar la información del entorno e intercambiarla entre los propios objetos o a través de Internet. Gracias a estas redes, es posible monitorizar cualquier métrica que podamos imaginar dentro de un edificio, y, después, utilizar dicha información para construir sistemas automatizados, conocidos como Sistemas de Gestión Energética para Edificios, capaces de extraer conclusiones sobre cómo utilizar de manera eficiente y óptima los recursos del edificio. Compañías pertenecientes a las TIC han previsto esta oportunidad de mercado que, en sincronía con la aparición de sensores más pequeños, eficientes y duraderos, permite el desarrollo de sistemas IoT eficientes. Sin embargo, la falta de acuerdo en cuanto a la estandarización de dichos sistemas está creando un escenario caótico, ya que se hace imposible la conectividad horizontal entre dichos sistemas. Además, la gran cantidad de datos a procesar requiere la utilización de técnicas de Big Data para poder garantizar respuestas en tiempos aceptables.

Esta tesis presenta, inicialmente, una arquitectura IoT estándar basada en la Nube que trata de hacer frente a los problemas anteriormente presentados mediante el uso de un middleware alojado en la Nube que ofusca la arquitectura hardware subyacente y permite la agregación de la información originada des de múltiples fuentes heterogéneas. Además, la información de los sensores se expone para que cualquier cliente de terceros pueda consultarla, después de haberse autenticado.

La utilización de sistemas IoT automatizados para manejar los recursos de los edificios requiere un alto nivel de fiabilidad, resistencia y disponibilidad. La pérdida de información no está permitida debido a las consecuencias negativas que podría suponer, como una mala toma de decisiones. Por eso, es obligatorio otorgar opciones de backup a las redes de sensores para garantizar su correcto funcionamiento incluso cuando se producen desconexiones parciales de la red. Adicionalmente, la colocación de los sensores dentro del edificio debe garantizar un consumo energético mínimo dentro de las restricciones de despliegue impuestas. En esta tesis, mejoramos el problema de colocación de los sensores para redes heterogéneas de sensores inalámbricos añadiendo restricciones de clustering o agrupamiento, para asegurar que cada tipo de sensor es capaz de obtener su métrica correspondiente, y restricciones de protección mediante la habilitación de rutas de transmisión secundarias. En cuanto a grandes redes homogéneas de sensores inalámbricos, esta tesis estudia aumentar su resiliencia mediante la identificación de los sensores más críticos.

Finalmente, presentamos un caso de uso de un Sistema de Gestión Energética para Edificios mediante una herramienta de simulación. Dicha herramienta utiliza como información de entrada modelos probabilísticos sobre las acciones de los ocupantes y modelos sobre la condición ambiental para actuar sobre los elementos del edificio y garantizar un funcionamiento óptimo y eficiente. Además, el comfort de los ocupantes también se considera como métrica a optimizar. Dada la imposibilidad de optimizar las dos métricas de manera conjunta, esta tesis también presenta un estudio sobre el *trade-off* que existe entre ellas.

Todo el trabajo presentado está pensado para otorgar ideas y herramientas para los sistemas IoT actuales y futuros, y asentar las bases para la estandarización que todavía está por llegar.

Resum

David Sembroiz Ausejo

Ambient Intelligence in buildings. Design and development of an interoperable Internet of Things platform

Durant molts anys, s'ha alertat a la població i als governs sobre l'increment en els nivells de pol·lució i d'emissió de gasos d'efecte hivernacle, que estan posant en perill la nostra vida a la Terra. El sector de les Tecnologies de la Informació i Comunicació, normalment conegut com les TIC, responsable de la informatització de la societat, ha estat senyalat com un dels sectors més importants encarregat d'agreujar tal problema. Però, molt esforç s'està posant per revertir aquesta situació mitjançant l'ús de recursos renovables, com l'energia eòlica o solar. Tot i que els governs han acordat seguir dit camí i evitar l'ús d'energia no renovable tant com sigui possible, no és suficient per erradicar el problema.

Encara que el sector de les TIC pugui semblar un problema afegit donada la gran quantitat i l'increment de dispositius connectats, les millores en tecnologia i en hardware estan habilitant noves maneres de lluitar contra l'escalfament global i l'emissió de gasos d'efecte hivernacle.

Durant les últimes dècades, companyies del sector públic i privat conscients del problema han centrat els seus esforços en la creació de solucions orientades a l'eficiència energètica tant a nivell de hardware com de software. Les noves xarxes troncals estan sent creades amb dispositius eficients i els proveïdors de serveis d'Internet tendeixen a crear sistemes conscients de l'energia per a la seva optimització dins dels seus dominis.

Seguint aquesta tendència, qualsevol nou sistema creat i afegit a la xarxa ha de garantir un cert nivell de consciència i un maneig òptim dels recursos que utilitza.

La informatització, anteriorment mencionada, ha forçat a les empreses a evolucionar el seu model de negoci cap a un més enfocat a la utilització de xarxes d'ordinadors per gestionar els seus recursos. Per això, dites companyies s'estan veient forçades a establir les seves seus centrals dintre d'edificis, per tenir un major control sobre la coordinació, connexió i maneig dels seus recursos. Això està provocant un augment en el consum energètic dels edificis, que s'estan convertint en un dels principals problemes.

Per poder fer front al problema, la Internet de les Coses o Internet of Things (IoT) ofereix nous paradigmes i alternatives per liderar el canvi. IoT es defineix com la xarxa d'objectes físics i virtuals, capaços de recol·lectar la informació per construir sistemes automatitzats, coneguts com a Sistemes de Gestió Energètica per Edificis, capaços d'extreure conclusions sobre com utilitzar de manera eficient i òptima els recursos de l'edifici. Companyies pertanyents a les

TIC han previst aquesta oportunitat de mercat que, en sincronia amb l'aparició de sensors més petits, eficients i duradors, permeten el desenvolupament de sistemes IoT eficients. Però, la falta d'acord en quant a l'estandardització de dits sistemes està creant un escenari caòtic, ja que s'està fent impossible la connectivitat horitzontal entre dits sistemes. A més, la gran quantitat de dades a processar requereix la utilització de tècniques de Big Data per poder garantir respostes en temps acceptables.

Aquesta tesi presenta, inicialment, una arquitectura IoT estàndard basada en la Neu, que tracta de fer front als problemes anteriorment presentats mitjançant l'us d'un middleware allotjat a la Neu que ofusca l'arquitectura hardware subjacent i permet l'agregació de la informació originada des de múltiples fonts heterogènies. A més, la informació dels sensors s'exposa perquè qualsevol client de tercers pugui consultar-la, després d'haver-se autenticat.

La utilització de sistemes IoT automatitzats per gestionar els recursos dels edificis requereix un alt nivell de fiabilitat, resistència i disponibilitat. La perduda d'informació no està permesa degut a les conseqüències negatives que podría suposar, com una mala presa de decisions. Per això, és obligatori atorgar opcions de backup a les xarxes de sensors per garantir un correcte funcionament inclús quan es produeixen desconnexi ons parcials de la xarxa. Addicionalment, la col·locació dels sensors dintre de l'edifici ha de garantir un consum energètic mínim dintre de les restriccions de desplegament imposades. En aquesta tesi, millorem el problema de col·locació dels sensors per xarxes heterogènies de sensors sense fils afegint restriccions de clustering o agrupament, per assegurar que cada tipus de sensor és capaç d'obtenir la seva mètrica corresponent, i restriccions de protecció mitjançant l'habilitació de rutes de transmissió secundàries. Pel que fa a grans xarxes homogènies de sensors sense fils, aquesta tesi estudia augmentar la seva resiliència mitjançant l'identificació dels sensors més crítics.

Finalment, presentem un cas d'ús d'un Sistema de Gestió Energètica per Edificis mitjançant una eina de simulació. Dita eina utilitza com informació d'entrada models probabilístics sobre les accions dels ocupants i models sobre la condició ambiental per actuar sobre els elements de l'edifici i garantir un funcionament òptim i eficient. A més, el confort dels ocupants també es considera com mètrica a optimitzar. Donada l'impossibilitat d'optimitzar les dues mètriques de manera conjunta, aquesta tesi també presenta un estudi sobre el *trade-off* que existeix entre elles.

Tot el treball presentat està pensat per otorgar idees i eines pels sistemes IoT actuals i futurs, i assentar les bases per l'estandardització que encara està per arribar.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors Davide Careglio and Sergio Ricciardi for their continuous support of my study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisors, I would like to also thank Josep Solé Pareta, Jordi Perelló Muntan and Jaume Comellas Colomé, for their insightful comments and encouragement to winden and improve the quality of this thesis.

I would also like to express my deepest gratitude to my mother Nati, and my sister Ainhoa. This dissertation would not have been possible without their continued support and patience.

Last but not least, I would like to especially thank my partner Iranzu, for all her love and endless patience. Without you, this thesis would have not been possible.

Contents

List of Figures

List of Tables

List of Abbreviations

6LoWPAN	IPv**6** over **Lo**w power **W**ireless **P**ersonal **A**rea **N**etworks
AmI	**Am**bient **I**ntelligence
AMQP	**A**dvanced **M**essage **Q**ueuing **P**rotocol
API	**A**pplication **P**rogram **I**nterface
BEMS	**B**uilding **E**nergy **M**anagement **S**ystem
BLE	**B**luetooth **L**ow **E**nergy
Bluetooth SIG	**Bluetooth S**pecial **I**nterest **G**roup
BMS	**B**uilding **M**anagement **S**ystem
BTU	**B**ritish **T**hermal **U**nit
CES	**C**onsumer **E**lectronics **S**how
CoAP	**Co**nstrained **A**pplication **P**rotocol
DDS	**D**ata **D**istribution **S**ervice
DTLS	**D**atagram **T**ransport **L**ayer **S**ecurity
GHG	**G**reen**h**ouse **G**as
GRASP	**G**reedy **R**andomised **A**daptive **S**earch **P**rocedure
HTTP	**H**yper**t**ext **T**ransport **P**rotocol
HVAC	**H**eating, **V**entilating and **A**ir **C**onditioning
ICT	**I**nformation and **C**ommunication **T**echnology
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IETF	**I**nternet **E**ngineering **T**ask **F**orce
ILP	**I**nteger **L**inear **P**rogram
IoT	**I**nternet **o**f **T**hings
ISP	**I**nternet **S**ervice **P**rovider
IT	**I**nformation **T**echnology
LoRaWAN	**Lo**ng **Ra**nge **W**ide **A**rea **N**etwork
LoS	**L**ine **o**f **S**ight
M2M	**M**achine to **M**achine
MIP	**M**ixed **I**nteger **P**rogramming
MIT	**M**assachustts **I**nstitute of **T**echnology
MQTT	**M**essage **Q**ueuing **T**elemetry **T**ransport
NFC	**N**ear **F**ield **C**ommunication
OTA	**O**ver **t**he **A**ir

PMV	**P**redicted **M**ean **V**ote
QoS	**Q**uality **o**f **S**ervice
RCL	**R**estricted **C**andidate **L**ist
RFID	**R**adio **F**requency **Id**entification
SASL	**S**imple **A**uthentication and **S**ecurity **L**ayer
SOA	**S**ervice **O**riented **A**rchitecture
TCP	**T**ransmission **C**ontrol **P**rotocol
TLS	**T**ransport **L**ayer **S**ecurity
UDP	**U**ser **D**atagram **P**rotocol
W3C	**W**orld **W**ide **W**eb **C**onsortium
Wh	**W**att-**h**our
WSAN	**W**ireless **S**ensor and **A**ctuator **N**etwork
WSN	**W**ireless **S**ensor **N**etwork

Dedicated to the memory of my father, who always believed in my ability to be successful in the academic arena. Your belief in me has made this journey possible.

Chapter 1

Introduction

The Information and Communication Technology sector, commonly known as the ICT sector, has been in the spotlight for many governments due to high carbon emissions which are endangering everyone's life by provoking unwanted changes to our World, such as temperature increment and high pollution levels in major cities.

The Earth's global mean temperature has been increasing at the fastest rate in record history, and the common agreement among the scientists is that such increment is mainly due to human-caused pollution. Focusing only on the ICT sector, BIO Intelligence Service released in 2012 a report which estimated that ICT sector use was contributing 2% of global GHG emissions. It also estimates that by 2020, this percentage will increase until 4%. More recent studies show the success of those predicted numbers. Authors in [1] estimate that, by 2020, global GHG emissions for the ICT sector will grow up to 3.6%. Additionally, the future estimation is not favourable, since it is expected that such GHG emissions contribute up to 14% of the total ones by 2040.

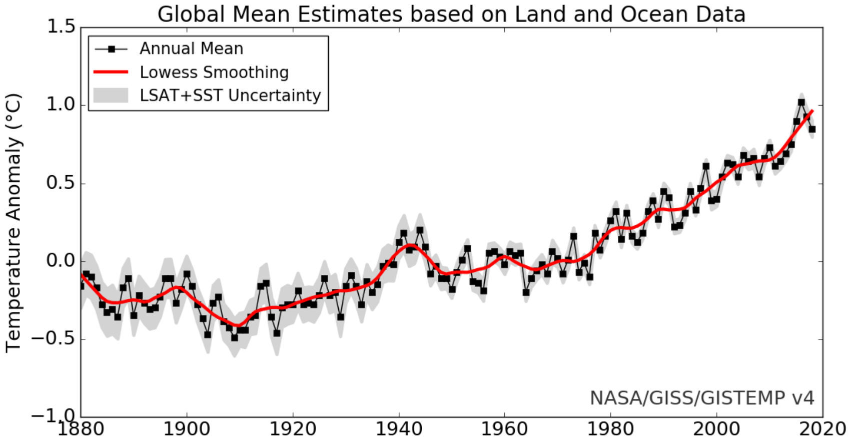


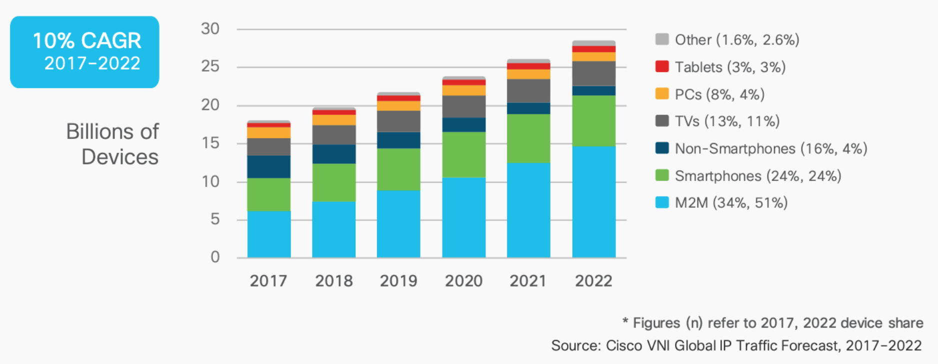
Figure 1.1: Global mean temperature estimates using ocean and land data [2].

This footprint, in addition to all the other contributing sectors, is endangering our lives on the planet due to the average temperature increment and the poles thaw.

As can be seen in Figure 1.1, since 1980, the global mean temperature has been increasing, and it is expected that such trend continues as previously stated. In order to reverse this situation, governments and major lobbies should take action, since individual action is not enough anymore and massive global changes must be carried out.

Many efforts have been put in order to lower the usage of non-renewable resources, mainly focused on the usage of renewable energies such as wind or solar power. For instance, renewable energy in Spain represented 42.8% of total electricity generation in 2014 according to the official Spanish Electric Report [3], and it is expected to switch its electricity system entirely to renewable sources by 2050 [4].

Although the ICT sector might seem an added problem to such scenario with 28.5 billion devices connected to the Internet by 2022, as CISCO states in [5] and depicted in Figure 1.2, new technologies and improvements can also enable new ways of fighting against global warming and GHG emissions.



* Figures (n) refer to 2017, 2022 device share
Source: Cisco VNI Global IP Traffic Forecast, 2017–2022

Figure 1.2: Connected devices forecast for CISCO between 2017 and 2022 [5].

During many years, researchers and enterprises aware of the current problem have focused their efforts on creating solutions for these problems in terms of energy-aware and energy-efficient products as well as algorithms and devices. Energy efficient devices such as routers have already been deployed in backbone networks and many ISPs tend to create energy-aware systems inside their domain. Regarding algorithms, many routing algorithms have been studied and presented to try to cope with the energy problem without disrupting user experience. Under this scenario, we developed a hybrid routing and wavelength assignment algorithm [6] that, when the network is lightly loaded, operates in an energy-efficient way, by routing the connections through the paths requiring the lowest amount of energy, while, when the network load increases, it dynamically switches to a pure load-balancing scheme in order to best allocate the available communication resources. The switching decision among load-balancing and energy-awareness is taken dynamically, driven by a threshold on the number of new connections requests reaching the network during a prefixed time window. An overview of such algorithm is described in Appendix B due to its indirect connection with this thesis.

Following this trend, it is currently infeasible the creation of a new part of the network without initially taking into account environmental impact.

Since the explosion of the new technologies thanks to their improvements, the computerisation of any company despite their size have aggravated the pollution problem. That is why buildings are now one of the most studied topic, since many companies are placed in shared buildings with shared systems such as Heating and Ventilation Air Conditioners. As an example of current building consumption data, the U.S. Energy Information Administration concluded that in 2018, 40% of U.S. total energy consumption was consumed in residential and commercial buildings, which amounts for about 40 quadrillion British Thermal Units [7, 8] or equivalently, 11.72 quadrillion Wh. Similarly, the European Commission stated that buildings are responsible for approximately 40% of energy consumption and 36% of emissions in the EU [9].

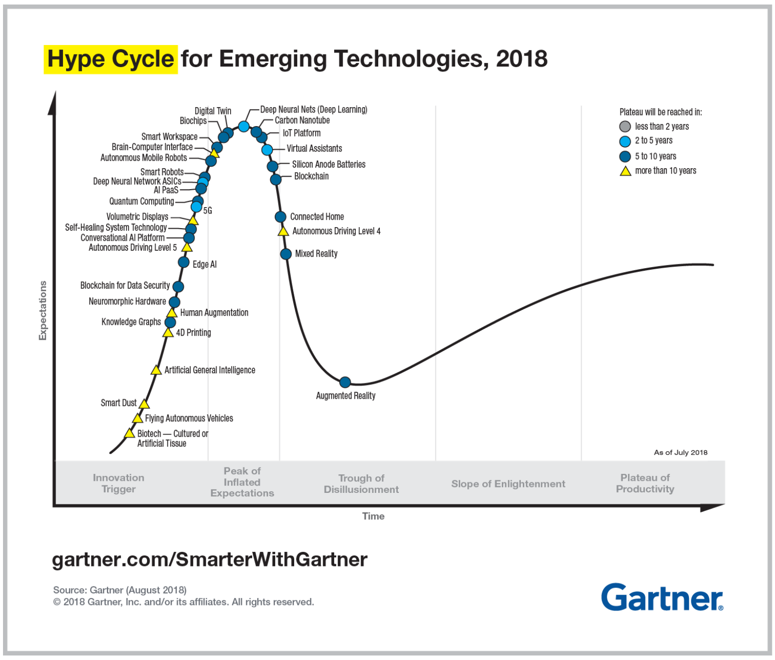


Figure 1.3: Gartner hype cycle for emerging technologies in 2018.

In order to cope with this problem, the Internet of Things (IoT) paradigm offers new alternatives for leading the change. IoT is commonly defined as the network of physical and virtual objects, devices or *things* that are capable of collecting surrounding data and exchanging it between them or through the Internet [10]. The term IoT was firstly used by Kevin Ashton, co-founder and executive director of the Auto-ID center at MIT, in 1999 [11], but for companies such as CISCO, the IoT was born in 2009, when more devices than people were connected to the Internet. At that time, the number of connected devices was 10 billion, but the expectations are generous. It is thought that by 2020, more than 50 billion devices will be connected to the Internet. Even though IoT is still a very new field as seen in Figure 1.3, and many research is needed in all its aspects, such as sensor specification, architecture definition, client-side applications, and human interaction, current technology improvements have led to a sustainable and worthy scenario in which devices are no longer research-related and can be seen as a legit product capable of offering a service.

1.1 Motivation

The motivations behind this thesis are directly related to the aggravated energy situation that our World is suffering. High GHG emissions due to the amount of fossil fuel utilised, and unsustainable peak consumptions are drastically reducing natural storages of raw material as well as increasing pollution. Even though technology can be seen as a twofold factor in this problem, its improvements create new opportunities to develop more sophisticated systems that are aware of their footprint, and, thus, optimise their consumption. Specifically, new improvements in IoT components are raising the possibility to monitor any kind of device or machine. Some of such technology improvements are the following:

Smaller, more durable and powerful sensors. The usage of smaller processors allow the overall reduction of the sensor size and also introduce durability elements to permit their positioning in delicate areas, to monitor and extract valuable information from dangerous scenarios for humans.

Increased efficiency. One of the key aspects of the Internet of Things paradigm is the wireless interconnection between devices. Thus, these devices must be equipped with autonomous power supplies that limit their lifespan. To cope with this problem, manufacturers are aiming for efficient processors and software engineers are specifically designing software and communication technologies for IoT in which lower energy consumption is the main requirement. To achieve this, sensors usually work in low-power-usage mode. Devices remain in sleep mode until a new sample message needs to be generated. Then, it wakes up, creates the message and transmits it by powering up the RF power amplifier. When the message is transmitted, both the RF power amplifier and the device are turned down until the next cycle.

Lower production cost. The improvements in industry and the easiness in which mass production is currently achieved allow companies to lower the price of each component.

The combination of these improvements created new market opportunities that companies have foreseen. Since IoT is in a very young state, the lack of coordination and the rapidness with which new gadgets are created are hampering the standardisation of the future Internet.

Architecture-wise, many companies have foreseen the importance that IoT will have in future cities, buildings and even cars. Because of this, they are creating and building individual and vertical architectures [12, 13] to provide the hardware and software needed to enhance our lives with smart capabilities thanks to IoT. Nevertheless, this scenario is not future-proof, due to the inability to interconnect devices and architectures from different manufacturers.

Focusing on buildings, many companies are putting their efforts into creating IoT platforms capable of collecting any kind of building data such as device power consumption, occupancy or device usage, among others. These data can later be deeply examined in order to better understand how each particular building is behaving and, thus, take action to create a more comfortable and efficient place. Buildings enhanced with IoT technology are commonly known as Smart Buildings, and the IoT system itself is usually known as Building Energy Management System (BEMS).

All of these improvements have led to the creation of a new research field known as Ambient Intelligence, in which buildings provide their data thanks to the sensors placed across the floors and reactive actions are taken thanks to Intelligent systems that receive the data and act according to predefined rules. These actions are meant to increase people's comfort and also to manage more efficiently the building resources without hampering the routine of the occupants.

Even though this scenario might seem sufficient, the energy consumed increases at a very fast pace and further research must be done to ensure better resource management and energy usage. Initial BEMS were thought as reactive systems that react to a change occurred in the building. However, nowadays it is possible to collect data from even personal devices to create particular behavioural patterns for every individual. This improvement allows the prediction of future actions and enables the possibility to create proactive Building Energy Management Systems capable, for instance, of adjusting room temperature before the occupant has arrived at the building, and, thus, increase his comfort whilst also utilising efficient device modes.

1.2 Internet of Things: A Key Solution

Even though IoT cannot directly address the climate change problem only by itself, it enables the possibility to automatically deal with the energy consumption in different situations which heavily contribute to high GHG emissions.

However, IoT networks are required to be Internet-compliant. This means that the backbone network remains intact and that IoT will take responsibility for the edge network. During many years, energy efficiency in both copper and optical networks has been a hot research topic and many solutions have been published, as previously stated.

By enhancing the most consuming elements with low-powered devices that open the possibility of an efficient resource management, it is possible to develop energy-aware systems and, thus, lower the ICT sector impact at least on the utility section. The most important and effective current solutions are:

Smart Healthcare Constantly collect patient data such as heart-rate, blood pressure, temperature, skin colour, etc. All the data is then aggregated into a centralised client that doctors can use to monitor all the patients and receive alerts if anything surpasses the normal intervals [14, 15]. Additionally, it is possible to monitor the physical position of every patient to optimise their movement and localisation in critical hospital sections such as emergencies. Similarly, the tracking of the patients can be further extended my monitoring their status and position during their daily life, and send position alerts when an abnormal situation is detected [16].

Smart Grid Collect electricity consumption data of a home by placing sensors at power meters, plugs and other devices connected to the utility. With such data, it is possible to create consumption patterns to predict future consumption rates to, for instance, generate and store green energy instead of utilising the electrical network [17]. Also, when the system detects no undergoing activity in the interior of the home, it can automatically turn *off* the power-hungry devices.

Smart Traffic Receive the status of the parking lots of a city in real-time to avoid congestion in city regions with no available parking space. Moreover, drivers save fuel consumption and travel time by directly going to the nearest available free parking lot [18].

Smart Buildings New internet-enabled thermostats or lightning systems can recognise behavioural and consumption patterns to automatically adjust indoor elements to the optimal state regarding energy optimisation and occupant comfort. By introducing smart elements into the buildings, people are more aware of their contribution to the overall problem [19, 20, 21].

Smart Cities Similarly to parking lots, by monitoring semaphores and traffic congestion, it is possible to extract traffic patterns to optimise the routing of the drivers and avoid potential congestions [22, 23].

As can be seen, IoT allows the improvement of several parts of the overall network. Smart Grids are capable of making the power supply chain more efficient by enabling improved monitoring and control, which means reductions in energy losses, greater network operational efficiency, better quality and reliability of energy supply, greater customer control of their energy use, etc. In the case of the edge network, smart parking lots, buildings, and cities allow its optimisation.

The possibility to attach small hardware to any kind of electronic or mechanic device enables the possibility to monitor everything. That is why the IoT is also defined as the interaction with *anything, anytime, anywhere.*

1.3 Thesis Objectives and Contributions

This section summarises the objectives of the thesis as well as the contributions. The objectives of the thesis can be divided into several IoT topics. Focusing on the architecture of IoT, the thesis presents an overview of the evolution that such architectures are suffering as well as a novel Cloud-based IoT architecture that tries to cope with some of the major architectural issues.

Data acquisition is carried over by Wireless Sensor Networks. Focusing on the characteristics of the sensors that comprise the network, WSNs can be divided into heterogeneous or homogeneous networks. For heterogeneous networks, we study the placement problem that appears due to the necessity to place each sensor at the optimal place in order to cover the desired sensing area and gather valuable information with minimal cost. Since new IoT systems are placed in already built factories or buildings, it is necessary to study the best plausible locations for sensor deployment in order to, then, select a subset of such locations to deploy the overall WSN and guarantee full coverage of the monitored elements and environment.

Furthermore, we study the criticality problem that appears in homogeneous WSN. That is, detect the most essential sensors of the network in order to provide backup capabilities to them and guarantee high reliability and availability, without wasting resources in creating backup plans for the rest of the network.

Then, data consumption in IoT is usually carried over by an end user or by a resource management application. In the case of end users, data can be offered by means of *wearables*, display panels placed in public areas or mobile applications, among others. Examples of such offered data can be health information in smart bracelets, traffic congestion in public parking lots, etc. Resource managers are, instead, software capable of monitoring all the elements inside a given scenario and inform when an abnormal situation appears. For instance, it is possible to monitor all the machines and environmental conditions inside a factory and alarm the security or management team when an specific metric reaches undesired values. Also, such managers can be more proactive and function, for instance, as autonomous systems for energy optimisation, without the need of human intervention.

1.3.1 Architecture Evolution

IoT architectures are proliferating and are being evolved at a very fast pace due to the early stages of their development and the lack of standardisation agreements. Initial IoT systems were thought as local systems with no Internet connection and few monitoring components. Also, data was consumed on the fly without any intermediate component for storage and processing. Connectivity between devices was also little or non-existent due to the lack of efficient transmission protocols and hardware.

As more effectiveness and efficiency was added to such IoT elements, architectures were also upgraded to a more global approach. Particularly, improvements in communication protocols allowed the interaction between devices and the creation of complex sensor networks for monitoring bigger area scenarios.

Moreover, the vast amount of data generated by the sensor network requires substantial storage capacity and processing power to be able to extract valuable information from it. To this aim, Cloud platforms appear as a crucial IoT component for storing, preprocessing and delivering data to IoT clients in very little time, anywhere.

Our contribution introduces, at its time, a novel Cloud-based IoT architecture, separated into several layers, that ease the interconnection of heterogeneous sensors and the delivery of sensor information to any IoT client, after the proper authentication. To summarize, the main features of the architecture are:

Underlying middleware Utilisation of an underlying middleware layer to receive heterogeneous sensor network data, pre-process it to create normalised messages and store them in the Cloud. The Cloud does not need to vary and extend its compatibility due to the heterogeneity management handled by such underlying layer.

Cloud-based Utilisation of Cloud storage for creating virtual sensors and storing sensor data. Additionally, Cloud technologies allow for rapid duplication and scalability to cover current network demand.

Shareability Exposure of Cloud virtual sensors to the public for allowing subscriptions and message update reception by any third-party application, after the proper authentication. Virtual sensors can either be public or private. Sensor identity and physical characteristics are stored in a public database.

The combination of an underlying middleware and a Cloud platform permits the architecture to obfuscate the physical characteristics of the hardware used as well as the heterogeneity of the WSN underneath.

As a result of this work, the following book chapter has been published:

> **Sembroiz, D.**, Ricciardi, S., and Careglio, D. "Chapter 10 - A Novel Cloud-Based IoT Architecture for Smart Building Automation". In: *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks.* Ed. by Massimo Ficco and Francesco Palmieri. Intelligent Data-Centric Systems. Academic Press, 2018, pp. 215 - 233. ISBN: 978-0-12-811373-8. DOI: 10.1016/b978-0-12-811373-8.00010-0.

This Cloud architecture is presented as a general tool that can be utilised by any IoT scenario. Furthermore, we implement and test this architecture in a building environment.

1.3.2 Sensor Placement

Sensor placement is a well-known problem heavily studied inside the field of Wireless Sensor Networks. However, the combination of such WSNs inside IoT requires new definitions and study in order to cover all the new requirements that may appear due to the IoT perspective of the solutions. The implementation of IoT in some locations require very high reliability to avoid disrupts in the network usage and data acquisition. Usually, such networks are formed by heterogeneous sensors that gather their surrounding information, and gateways, which are more powerful devices with connectivity capabilities, that aggregate the data and retransmit it to the sink node. The sink node is the central node of a WSN, and is responsible for sending all the aggregated data to either an application endpoint or a Cloud middleware that accumulates and exposes the data. One may argue that the sink node is the most important one inside a WSN, because its failure completely disables the forwarding of all the WSN data. However, within an IoT perspective, all the sensors inside a network can be important, and, depending on the solution under deployment, mandatory. A factory relying on sensors to acquire the status of critical machines cannot allow fault states.

Additionally, since new IoT systems are being deployed in already built buildings, factories, or even cities, it is needed to decide the optimal locations for the placement of such sensors in order to fully cover the area under study with minimal cost. To do so, we present an Integer Linear Program (ILP) capable of selecting the optimal locations from a set of given plausible ones, for both sensors and gateways. Also, since the variables and elements that need to be monitored usually require heterogeneous sensors, it is needed to restrict the positions in which every type of sensor can be placed.

To cope with the aforementioned disruptance problem, the ILP also introduces a *protection level* reliability requirement. The *protection level* ensures that, in the case of a partial network disruptance, every sensor can reconsider their transmitting path in order to avoid such disruptance and, thus, guarantee that their data is not lost. Depending on the criticality of the WSN under study, a low *protection level* might not be enough, and additional levels can be needed in order to increase the availability percentage of the network. However, the increase of the *protection level* presents a trade-off with the optimality in terms of network consumption and deployment cost. We also present a study to identify the increase in network consumption as the *protection level* increases for the case of a specific building layout.

Results for the ILP and the *protection level* study are presented in the following journal article:

> **Sembroiz, D.**, Careglio, D., Ricciardi, S., and Fiore, U. "Planning and operational energy optimisation solutions for smart buildings". In: *Information Sciences* 476, pp. 439–452. ISSN: 0020-0255. Current Impact factor: 5.524. DOI: 10.1016/j.ins.2018.06.003.

1.3.3 Sensor Criticality Detection

Sensor criticality is another topic widely studied in the WSN field. In this thesis, we study the criticality problem for multi-hop homogeneous WSNs. Literature defines the *critical* node of a WSN as the node that, once disrupted, disconnects the network into two separate set of nodes, which cannot communicate due to the lack of transmission paths.

For our contribution and study, we have defined the *criticality* of a node as the percentage of disruptance its failure adds to the network, without completely disconnecting it. It can be seen that, with this definition, nodes can be ranked per their criticality, instead of only selecting the *critical* node that disconnects the network. For this study, the *criticality* of the sensors is calculated by checking two of the main WSN quality metrics, namely latency and lifetime. Latency makes reference to the maximum *time* needed for a package to reach the sink node, and lifetime is calculated as the amount of time the network is fully operational, i.e., until the first node runs out of battery and it disrupts the network performance.

Due to the possibility to pinpoint a set of critical nodes instead of the most critical one, the problem becomes infeasible for programs delivering the exact solution when the size of the input testbed is very large. Commonly, such problems are solved by means of ILPs capable of accurately identifying optimal solutions. However, as the size of the problem or the size of the required solution increases, programs require large computational power and execution time. To cope with this problem, we present a GRASP meta-heuristic capable of detecting up to the 5 most critical nodes inside a large network. The usage of a meta-heuristic, however, has the hindrance of not always detecting the optimal solution. To solve this, we also present a set of *features* related to sensor nodes, namely connectivity, relay and hops to the base station, that can be calculated prior to execute the GRASP meta-heuristic. The *features* permit the creation of a criticality ranking which allows us to prune the set of inputs to be tested in order to obtain the optimal solution.

To validate such *features*, we show a comparison between our GRASP meta-heuristic results and the ILP ones presented in the literature for small networks, i.e., 100 sensors distributed in a $200m$ radius area. Once the *features* are validated, they are used for larger networks in order to study latency and lifetime increase as the network size increases.

In particular, we consider networks up to 1000 sensor nodes, distributed by using two different techniques. Firstly, the initial network previously mentioned is scaled up maintaining node density. That is, as the number of nodes increases, the radius area under deployment also increases. Then, the initial network is scaled up by fixing the $200m$ radius deployment area in order to increase node density.

The definition of the study and the results have culminated in the following journal article:

> **Sembroiz, D.**, Ojaghi, B., Careglio, D., and Ricciardi, S. "A GRASP Meta-Heuristic for Evaluating the Latency and Lifetime Impact of Critical Nodes in Large Wireless Sensor Networks". In: *Applied Sciences* 9.21 (2019). ISSN: 2076-3417. Current Impact factor: 2.217. DOI: 10.3390/app9214564.

1.3.4 Building Management System

As previously stated, the possibilities that IoT enables are endless. In particular, IoT is a new tool capable of tackling the daily energy consumption problem present in our lives. The possibility to monitor any kind of device and extract information about its consumption and usage allows the development of fine-grained functioning patterns. These patterns can then be studied in order to optimise the behaviour of such devices. Optimisations can be done in regard to multiple parameters, but the most common ones are energy consumption and user comfort. The combination of exhaust monitoring and the possibility to automatically change the status of a device thanks to actuators enable the possibility to reduce wasteful and inefficient scenarios. For instance, the drivers of a factory can be guided by an automatic system that gathers facility information and avoids congestion whilst optimising the resources.

Under this scenario and specifically for buildings, BMS appeared to integrate the management and monitoring of all building devices into one single piece of software. A BMS can be defined as a software tool that aggregates the information of all the sensors deployed under a certain area, interprets the data and extracts valuable information for configuring the building automatically and optimally. Thus, human intervention is drastically reduced and buildings become more energy-friendly. In addition to that, it is also possible to measure different metrics, such as occupant comfort, and configure the BMS to take into consideration any given number of metrics. However, depending on the relationship between those metrics, it is usually not possible to simultaneously optimise all of them. For instance, a BMS can control indoor environmental conditions by collecting temperature, humidity, light levels, etc. and actuate over windows, doors, HVACs, and lights accordingly to reduce building energy consumption and also increase occupant comfort.

In this thesis, we present a BEMS as a simulation tool capable of evolving the status of a given building layout, collect the energy and comfort values at any given moment of all the indoor elements, and act accordingly depending on the metrics under optimisation. The high cost of a real testbed and the security flaws that it might have due to privacy concerns led us to utilise a simulation software instead.

In order to create a close-to-reality simulation scenario, environmental profiles for temperature, humidity, and luminosity are created. Moreover, user profiles are individually specified depending on the role of the different occupants. The actions that emulate the movement of the occupants inside the building are modelled as probability distributions over time. Furthermore, we introduce a *prediction threshold* parameter that stipulates the time window with which the BEMS is capable of anticipating future actions.

With all of this, our BEMS aims at reducing overall energy consumption while fulfilling the predefined occupants' comfort constraints, such as desired temperature or luminosity. Since the two metrics under consideration are not directly proportional, we also study the trade-off that exists between them in order to show whether it is possible to maintain proper energy consumption levels and high comfort rate. The study is done taking into consideration different *prediction threshold* values.

Results of the comparison between energy consumption and occupant comfort for different prediction thresholds have been published in the following journal article:

> **Sembroiz, D.**, Careglio, D., Ricciardi, S., and Fiore, U. "Planning and operational energy optimisation solutions for smart buildings". In: *Information Sciences* 476, pp. 439–452. ISSN: 0020-0255. Current Impact factor: 5.524. DOI: 10.1016/j.ins.2018.06.003.

All the aforementioned calculations and results have been extracted using a MacBook Pro early 2015 (Apple®, Cupertino, CA, USA) with 2.7 GHz Intel® Core^{TM} i5 processor (Santa Clara, CA, USA) and 8 GB 1,867 MHz DDR3 memory.

Chapter 2

Related Work

Preface

This chapter collects the literature regarding IoT. Due to the high amount of fields that IoT comprises, literature has been divided into sections, each of which corresponds to a specific problem present in new IoT systems. Initially, physical and application protocols are listed in order to compare their specifications. Then, literature for the planning problem for both homogeneous and heterogeneous WSN is presented. Finally, literature for the operational phase is cited. Note that the operational phase is focused on the efficiency and optimisation of a building through a Building Energy Management System.

2.1 Introduction

IoT systems are commonly divided into layers that physically correspond to the elements it encompasses. For instance, the lowest layer, namely the perception layer, is responsible for data acquisition via a WSN formed by sensors, actuators, and gateways. Similarly, data storage is carried over by repositories that gather all the underlying information and store it in a normalised form for future utilisation. This task corresponds to the middleware layer, which can either be local or global, with the aid of Cloud infrastructures.

Finally, such stored data is commonly utilised by the application layer, formed by client software that connects to the middleware in order to acquire the necessary data for extracting valuable information. Such information is, then, utilised for the automation and optimisation of a given scenario.

As can be seen, the aggregation of the aforementioned layers comprises the creation of a defined architecture framework. Due to the importance of intra-layer and inter-layer communication inside such architecture, Section 2.2 starts by presenting a background of the principal and most important protocols for transmitting data between elements inside the perception layer. Then, it presents the common protocols for connecting between the middleware and the application layers.

After that, Section 2.3 presents the literature regarding the optimisation of WSN with respect to different types of network and optimisation criteria. Specifically, Section 2.3.1

tackles the localisation and placement problem that appears in heterogeneous WSN, i.e., networks of sensors with distinct hardware and communication protocols. Then, Section 2.3.2 studies the node criticality detection problem of homogeneous WSN, i.e., networks with equal sensors.

To finalize, Section 2.4 lists the literature regarding Building Energy Management Systems and the optimisation of building parameters such as energy consumption and occupant comfort.

2.2 Architecture Background

This section presents a background of IoT enabling technologies and protocols used for the standardisation and definition of current IoT systems.

Since the beginning of the current Internet, many groups have been created for helping in the standardisation of protocols and technologies. Internet Engineering Task Force (IETF), World Wide Web Consortium (W3C), Institute of Electrical and Electronics Engineers (IEEE) are some of the most important ones.

In the initial steps of IoT, protocols such as RFID and NFC were the standard the facto mainly due to its low production cost. However, its transmission limitations in terms of range coverage and inability to communicate through the Internet hampers their usage in new IoT scenarios such as smart buildings or cities. In the industry sector, they are still widely used for packet tracking and object identification.

During the last years, groups have put their efforts into creating standards for protocols directly related to the Internet of Things. Even though it is possible to use old communication protocols such as Bluetooth or Wi-Fi, their characteristics do not cover new IoT device requirements. Many IoT devices rely on the necessity of having external power supplies, such as batteries, to work, requiring reduced power consumption and cost while maintaining a similar communication range with respect to their analogous current Internet protocols. To cite some, Bluetooth Low Energy, Wi-Fi HaLow or LoRaWAN are IoT focused protocols with the stated requirements fulfilled. Even though the core of such protocols is very similar, depending on the kind of application being developed, one may fit best than another. Moreover, the usage of multiple protocols inside the same system is not forbidden. For instance, a general system combining many information sources, each of them using the specific sensor devices, can use the protocol that fits best by taking into account device connectivity and location.

IoT enabling protocols can be divided into two major groups, namely Infrastructure protocols and Application protocols. Infrastructure protocols refer to the ones that actuate inside the underlying infrastructure and create the communication between the data acquisition stage and the storage stage of the system. For instance, the interconnection between sensors inside the perception layer or the connection between the perception layer and the network layer. Application protocols are responsible for interconnecting the infrastructure with the application.

In the following sections, the most relevant protocols of both groups are explained more in depth. To summarize, a table comparing the relevant characteristics and parameters is also shown.

2.2.1 Wireless Infrastructure Protocols

Radio Frequency Identification (RFID)

The Radio Frequency Identification protocol can be seen as the first protocol utilised inside IoT systems. RFID devices are wireless microchips or tags attached to any kind of element without intelligence for its automated identification and tracking. RFID is very used in industry, due to the ability to communicate with the tags without being in their line-of-sight. The tags can either be passives, actives or battery-assisted passive.

Passive tags are the cheapest and smallest ones due to their simplicity and lack of on-board batteries. Such tags need the presence of an active reader for receiving the radio energy transmitted by the reader.

Battery-assisted tags contain small batteries inside and are activated in the presence of an active reader. Once active, tags transmit their signals.

Active tags contain on-board batteries and work proactively by continuously transmitting their ID signals without the need of an active reader. Once a reader enters the range signal, it automatically receives the data.

The uses of RFID tags are many among different sectors, such as access management, inventory count, people and animal tracking, airport baggage tracking logistics, among others.

RFID tags and readers can work in ultra-high (850 - 960 MHz), high or low frequency, and the transmission range and speed varies depending on the frequency under use. Low-frequency RFID typically operates at 125 - 134 KHz, and delivers a short read range of 10cm. Transmission speed is also slow compared to higher frequencies. High-frequency RFID operates at 13.56 MHz with reading ranges between 10 cm and 1 m. Ultra-high frequency uses the 860 - 960 MHz band, but most countries operate between 900 and 915 MHz. The UHF range can be as long as 12 metres and it has the fastest data transmission rate among all the aforementioned frequencies.

The optimal frequency needs to be accurately selected depending on the system under development.

Near Field Communication (NFC)

Near Field Communication protocol enables simple and safe communications between electronic devices. However, the short operational range makes it limited for multiple IoT scenarios. The main usage of NFC inside IoT is the secure identification of transactions and accesses throughout the facilities of a city. NFC can be equipped in identification or payment cards, booking tickets or physical access elements to securely check the identity of the person performing the action without human intervention.

Moreover, due to the common implementation of NFC chips inside smartphones, those are being converted into centralised IoT sensors capable of performing all the aforementioned actions by storing digitalised versions of identity and payment cards, keys, etc.

NFC operates, at maximum, within a 4cm range. It utilises the 13.56 MHz RF carrier frequency and the data rate is also limited to 424 Kbps.

NFC can operate in three different modes, namely card emulation, reader/writer and peer to peer. In the card emulation mode, an active NFC reads the information from a passive device. For instance, a tag reader reading tickets to allow the entrance of people to a location.

During the reader/writer mode, one of the communication devices only exposes its information to be read, and the writer stores such information. Examples of reader/writer mode can be device pairing or smart advertising.

In the peer to peer or point to point mode, both devices send and receive data to establish a connection and interchange information. Examples of such mode are social media applications, direct smartphone to smartphone communication or automotive communication between intelligent cars to avoid collisions.

Bluetooth Low Energy (BLE)

Bluetooth Low Energy or Bluetooth Smart, as it has been branded, is an enhancement of the Bluetooth technology in which connectivity and power usage are smarter than its predecessor. However, devices with Bluetooth Smart technology attached are not compatible with previous versions. To cope with this problem, Bluetooth Special Interest Group completed the Bluetooth Core Specification version 4.0 to include compatibility between versions. Current devices include this new core protocol making them able to communicate with any Bluetooth device.

The shifting in the connection paradigm performed by the Internet of Things has forced new protocols to include new behavioural modes. Bluetooth Smart includes ultra-low peak, average and idle modes. Once the pairing between two devices is performed, Bluetooth Smart focuses on sending small bits of data when needed and putting the connection in a low power consumption mode in order to drastically reduce energy usage.

According to the Bluetooth SIG specification, this protocol has been specifically designed for smart home, health, sports and fitness sectors. These sectors can take advantage of the following Bluetooth Smart features [24]:

Power Low power requirements, allowing the devices to operate for months or even years.

Size Small size and low cost.

Compatibility Extensive compatibility with a large base of mobile phones, tablets, and computers, allowing the interoperability between such devices.

As for its technical details, Bluetooth Smart operates in the same spectrum range as its predecessor, the 2.4 GHz-2.4835 ISM band. However, the set of channels used vary significantly. Instead of the classic 79 1-MHz channels, Bluetooth Smart uses 40 2-MHz channels. Regarding bit rate and maximum transmission power, they are limited to 1 Mbit per second and 10 milliwatts respectively. Its range coverage is ten times that of the classic Bluetooth (10 metres versus 100 metres approximately). Latency wise, it is 16 times shorter (100ms versus 6ms) [25].

	Bluetooth Classic	**Bluetooth Smart**
Spectrum Range	2.4-2.4835 GHz	2.4-2.4835 GHz
Channel Bandwidth	1 MHz	2 MHz
Number of channels	79	40
Max. Bit Rate	3 Mbps	1 Mbps
Max. Transmission Power	100 mW	10 mW
Avg. Range	10 m	100 m
Avg. Latency	100 ms	6 ms

TABLE 2.1: Comparison between Bluetooth Classic and BLE [25].

ZigBee

ZigBee is a standard based on the IEEE 802.15.4 specification specially targeted for long battery life devices in wireless mesh networks. This protocol has been evolving since its appearance in 1999, and its last specification is called ZigBee PRO, from 2007. Even though it shares features with Bluetooth, ZigBee is intended to be simpler and less expensive.

Regarding operation bands, ZigBee uses the same as Bluetooth [26], the 2.4 GHz band. In some locations, this band varies. For instance, China uses the 784 MHz band, Europe uses the 868 MHz band, whilst the USA and Australia use the 915 MHz one.

Its simplicity also limits some important aspects such as the transmission rate and communication range. Unlike Bluetooth, data transmission is limited to a maximum of 250 Kbit per second, which may be enough depending on the scenario under use. The communication range varies between 10 and 20 metres for indoor transmissions, depending on power output and environmental characteristics.

6LoWPAN

The IPv6 over Low power Wireless Personal Area Networks or 6LoWPAN was created by a concluded working group in the Internet area of the IETF to fulfill the necessity to allow any kind of devices, even the smallest ones with limited power usage and processing capabilities, to participate in the Internet of Things [27].

6LoWPAN is a combination of IEEE 802.15.4 and IP in a simple well understood way. The key features of this protocol are the encapsulation definition and header compression that allow the compatibility between local area networks and wide area networks with IEEE 802.15.4 based networks.

Since 6LoWPAN pertains to the network layer of the OSI model, it does not have a specific transmission specification. Instead, the underlying link layer protocol is responsible for providing them. As mentioned before, this protocol has been designed to work on top of IEEE 802.15.4 based networks which provides the transmission characteristics already explained in Section 2.2.1.

Wi-Fi HaLow

Wi-Fi HaLow is a very new technology presented in January 2016 in the Consumer Electronics Show (CES) by the Wi-Fi Alliance [28]. This new Wi-Fi specification is directly suited to meet the unique requirements of IoT environments such as Smart Homes, Smart Cities and Industrial markets. It extends Wi-Fi, and specifically its 802.11ah specification, to operate into the 900 MHz band, enabling the low power connectivity necessary for applications including sensors and wearables which hardly rely on battery lifetime. Its range has been extended to almost twice that of current Wi-Fi, and will not only be capable of transmitting further but also providing a more robust connection in harsh environments thanks to its ability to penetrate walls or other barriers more easily.

Devices with HaLow support are expected to also support current 2.4 and 5 GHz Wi-Fi bands, allowing interoperability between current devices and new ones. They also support IP-based connectivity to natively connect to the Internet. Another important point worth mentioning is the ability to connect thousands of devices to a single access point to create dense device deployments. As for its transmission power, HaLow is expected to work between 150 Kbps and 18 Mbps depending on the requirements of the application. To support such transmission rates, different channel setups are required: 150 Kbps only requires a 1 MHz channel but the maximum transmission rate requires a 4 MHz-wide channel.

This new technology is the answer to Bluetooth, and Wi-Fi Alliance is certifying HaLow products since 2018.

LoRaWAN

LoRaWAN [29] is a Low Power Wide Area Network created by the LoRa Alliance as a solution for wireless battery-operated devices. It specifically targets the IoT main requirements such as secure communication, mobility and localisation services. In a typical LoRaWAN network, devices and gateways compose a star of stars topology in which only the gateways are connected to the Internet, whereas devices use single-hop wireless communication to transmit their data to single or multiple gateways simultaneously. The transmission between devices and gateways is bi-directional, but it also enables the possibility for multi-cast messaging for Over The Air software upgrade.

LoRaWAN supports a wide range of frequency channels and data rates. Moreover, transmission with different specifications to the same gateway does not interfere with each other. Every transmission is encapsulated in a separate *virtual* channel which increases the capacity of the gateway significantly. Data rates range between 0.25 Kbps and 50 Kbps.

LoRaWan defines three classes for endpoint devices to address the different needs reflected in the wide range of possible applications:

Class A Bi-directional end devices Asynchronous transmissions in which every uplink message is followed by 2 short downlink windows that the gateway can take advantage of to send messages to the end devices. After these windows have finished, the end devices are set to idle until the next uplink transmission. This class operates in the lowest power and it is suitable for applications that only need end devices to gateway communication.

Class B bi-directional end device with scheduled receive slots In addition to Class A random receive windows, end devices are told by means of a time-synchronised Beacon from the gateway which time slot they must listen for any possible downlink communication.

Class C bi-directional end devices with maximal receive slots End devices are continuously listening for downlink messages and this window is only closed when transmitting to the gateway. This class is usually targeted for AC-powered devices because of its high power consumption.

Table 2.2 acts as a summary and comparison between the main infrastructure protocols presented. To clarify, the communication range shows the distance to which these technologies can transmit depending if the transmitter and the receiver are in Line of Sight (LoS) or not. Regarding spectrum usage, it can vary depending on the location since the legislation varies in every continent.

	RFID	**NFC**	**BLE**	**ZigBee**	**6LoWPAN**	**Wi-Fi HaLow**	**LoRaWAN (EU)**
Spectrum Range	125/134 KHz, 13.56 MHz	13.56 MHz	2.4-2.4835 GHz	2.4-2.4835 GHz	868/915/2400 MHz	900 MHz	868 MHz
Bit Rate	4 - 640 Kbps	106 - 424 Kbps	1 Mbps	20 - 250 Kbps	250 Kbps	150 Kbps - 18 Mbps	0.25 - 50 Kbps
Peak Consumption	< 100 mA	40 mA	< 15 mA	30 - 40 mA	< 15 mA	∼ 50 mA	∼ 38 mA
Range	1 cm - 12 m	4 cm	10 - 100 m	10 - 100 m	10 - 200 m	1 km	2 - 22 km

Table 2.2: Comparison between the main enabling infrastructure IoT protocols.

2.2.2 Application Layer Protocols

Hypertext Transfer Protocol (HTTP)

The Hypertext Transfer Protocol (HTTP) is an application layer protocol designed for distributed, collaborative, hypermedia information systems. This protocol is the foundation of data communication for the World Wide Web.

HTTP was initiated in 1989 at the European Organisation for Nuclear Research (CERN). However, the development of standards was coordinated by the IETF and the World Wide Web Consortium (W3C), culminating in the publication of a group of RFCs in 1997, with the definition of the HTTP/1.1 version in [30] firstly, and updated in [31]. During many years this has been the standard *de facto*. In 2015, the successor HTTP/2 was standardised [32].

HTTP works as a request-response protocol in the client-server computing model. The majority of the time, it uses TCP as a transport protocol for reliability. However, it can be adopted to use unreliable protocols such as UDP.

Even though its usage has been extended to the IoT world, it was not specifically designed for this purpose. If compared to other IoT-oriented protocols, HTTP may not be the best choice due to its protocol overheads and communication requirements. However, it has served as a strong base for newly developed protocols such as CoAP.

Constrained Application Protocol (CoAP)

The Constrained Application Protocol (CoAP), shown in Figure 2.1, is defined as a *specialised web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things* [33]. As it can be extracted from the definition, this protocol is specifically tailored for the IoT and M2M applications. The major standardisation of this protocol has been carried out by the IETF Constrained RESTful environments (CoRe) Working Group and the core is specified in [34]. This application layer protocol can be seen as an enhancement of HTTP for low power devices. It is based on the successful REST model, in which resources are available under a URL and clients can access those resources using the GET, PUT, POST and DELETE methods. Additionally, CoAP also supports publish-subscribe thanks to the usage of an extended GET method.

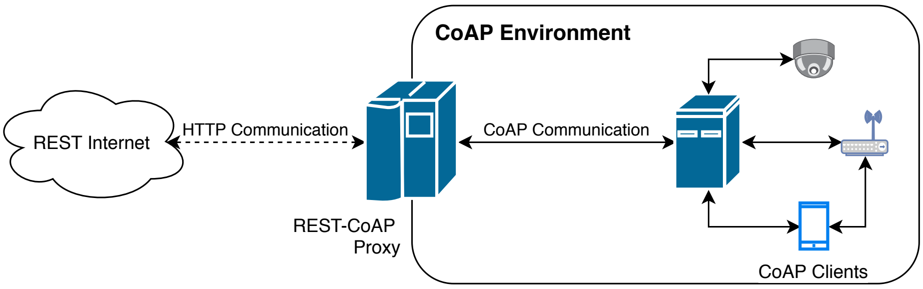


FIGURE 2.1: Constrained Application Protocol specification.

Even though it shares similarities with HTTP, CoAP is specifically designed to run over UDP only. As UDP is inherently not reliable, CoAP defines two types of messages, namely *confirmable messages* and *non-confirmable messages* to define its own reliability mechanism. The former requires an acknowledgement similar to the ACK used in TCP communications whilst the latter does not require any kind of acknowledgement.

Message Queue Telemetry Transport (MQTT)

Message Queue Telemetry Transport (MQTT), depicted in Figure 2.2, is a client-server publish-subscribe messaging transport protocol standardised under the ISO/IEC PRF 20922 [35]. It is lightweight, simple and very easy to implement. Its lightness characteristic makes it ideal for environments in which communication capabilities are limited such as M2M or IoT scenarios. Unlike CoAP, MQTT has been designed to run over TCP/IP or other network protocols that provide ordered, lossless and bi-directional communication. In this regard, MQTT is similar to HTTP. However, the former has been designed to have less protocol overhead.

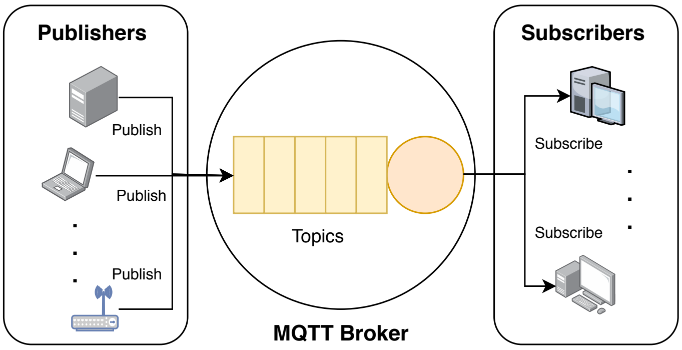


FIGURE 2.2: Message Queue Telemetry Transport protocol specification.

The reliability of messages in MQTT is taken care of by three Quality of Service (QoS) levels:

At most once Messages are delivered in a best-effort manner and messages loss can occur. This QoS is tailored for scenarios in which the loss of a message is not relevant.

At least once Messages are assured to arrive, but duplicates can occur.

Exactly once Messages arrive exactly one time, without duplicates. This QoS is reserved for systems that must operate reliably all the time, such as banking systems.

Thanks to the publish/subscribe model, it also allows for one-to-many message distribution with application decoupling.

Advanced Message Queuing Protocol (AMQP)

Advanced Message Queuing Protocol, represented in Figure 2.3, is an open standard application layer protocol for message-oriented middlewares. The defining features are message orientation, queuing, publish-subscribe and point-to-point routing, reliability and security, based on SASL or TLS. It provides a wide range of features related to messaging, such as reliable queuing, topic-based publish-subscribe, flexible routing, transactions, and security. As its name indicates, this advanced protocol permits a lot of fine-grained control over the queues and messages. Equally to MQTT, it also provides the message-delivery guarantees *at most once*, *at least once* and *exactly once* previously explained.

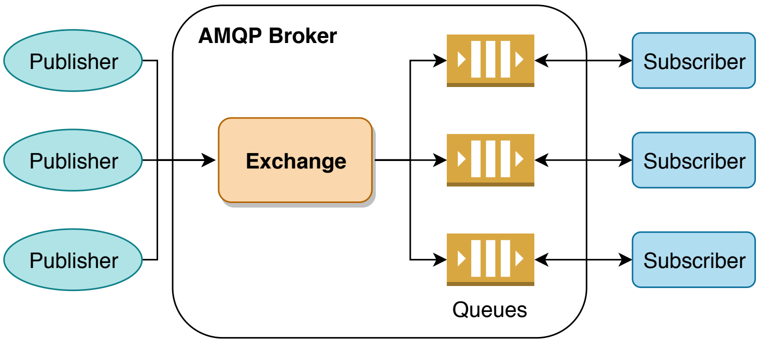


FIGURE 2.3: Advanced Message Queuing Protocol specification.

AMQP was designed for interoperability between different vendors and, thus, messages contain more overhead than its MQTT counterpart. For this reason, it was not initially designed to work with lightweight devices like IoT sensors. However, in the need of reliable message queue, AMQP can also work with such devices at the expense of additional energy consumption or storage.

Data Distribution Service (DDS)

The Data Distribution Service protocol, shown in Figure 2.4, is very similar to the MQTT protocol. Equally, it enables the interchanging of information thanks to the publish-subscribe pattern. However, instead of requiring a centralised broker that handles the management of the topics and the reception of the data, it utilises a decentralised Cloud capable of receiving and storing a large amount of messages.

DDS is responsible for handling message addressing, delivery, flow control, retries, etc. The nodes inside a DDS system can either be publishers, subscribers or both simultaneously.

Thanks to the decoupled architecture, DDS supports mechanisms that go beyond the basic publish-subscribe model. In particular, applications never need information about the other participating applications, including their existence of physical locations. DDS transparently determines the applications that should receive every message, the location of such receivers and the response in case of failure.

Moreover, DDS allows users to specify multiple QoS parameters to configure behavioural patterns and discovery.

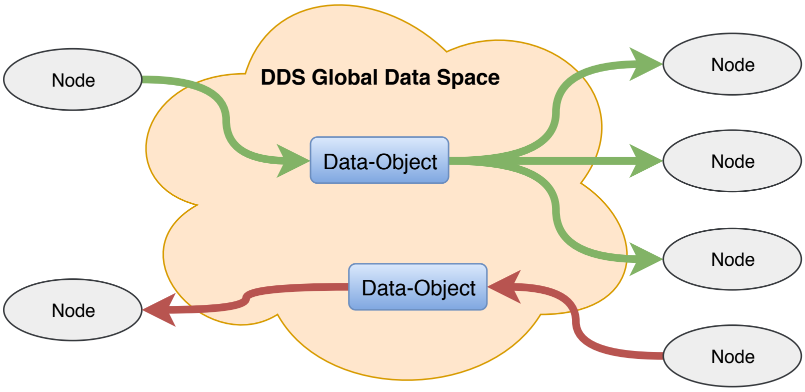


FIGURE 2.4: Data Distribution Service protocol specification.

Table 2.3 summarises the main specifications of the application protocols previously explained.

	HTTP	**CoAP**	**MQTT**	**AMQP**	**DDS**
Main Transport Protocol	TCP	UDP	TCP	TCP	TCP/UDP
Publish/Subscribe	✗	✓	✓	✓	✓
Request/Reply	✓	✓	✗	✗	✓
QoS	✗	✓	✓	✓	✓
Security		DTLS	TLS	TLS	TLS/DTLS/DDS Security
Fault Tolerance		Decentralised	Single Point of Failure	Implementation Specific	Decentralised

TABLE 2.3: Comparison between the main enabling application IoT protocols.

Focusing on the application protocols that allow publish-subscribe, it can be seen that the rest of the characteristics varies due to the design decisions. However, it is needed to consider all the protocols depending on the application under development. For large scale distributed applications, DDS can be a good candidate due to the decentralised nature and the security it enables. Conversely, small applications with limited resources and small operational range, MQTT might be a good choice due to its lightness.

2.3 Wireless Sensor Network Optimisation

The problem of optimising a WSN can be tackled from many different perspectives, depending on the objectives to optimise and the requirements, such as, e.g., topological restrictions, battery constraints, QoS requirements, and levels of resilience and security. Regarding the optimisation objectives, some problems are related to reducing the initial WSN deployment cost by minimising the number of nodes to install, while others focus more on WSN performance, by trying to minimise the overall network latency, maximize network connectivity to increase resilience in the event of unexpected failures, or minimise network energy consumption (especially in scenarios with constrained devices) [36].

Literature regarding WSN has been divided into two major sections. Section 2.3.1 presents the work related to the heterogeneous placement problem, i.e., the optimal placement of sensors and gateways in a given area for sensing all the required variables and components, from an IoT perspective. Section 2.3.2 describes the literature regarding the identification and degradation study of critical nodes inside an homogeneous WSN.

2.3.1 Sensor Placement

With the appearance of IoT, WSNs have gained popularity due to the necessity to monitor every physical device in a given environment. Such requirement has brought WSNs into a more heterogeneous perspective, since the characteristics of sensors may vary depending on the features being monitored.

Additionally, new scenarios are no longer restricted to floors or 2D locations. Instead, IoT enables the possibility to monitor 3D environments such as office buildings, hospitals, cities, etc. Even though our research can handle both 2D/3D scenarios due to the chosen modelling and transmission assumptions, a more accurate 3D model which also takes into account line of sight as in [37] would be necessary.

The utilisation of heterogeneous sensors enriches the optimisation problem by requiring proper positioning of each type of sensor in a specific region, instead of only choosing the positions in which sensors must be placed. This problem can be seen as a clustering problem in which every cluster region must contain a predefined set of sensors. Authors in [38] present an initial approach to such problem, restricting the positioning of every type of sensor into a subsection of the overall layout. However, the authors do not consider the possibility to overlap clusters of different sizes and, thus, the possibility to reduce the number of required set of sensors.

In addition to sensor placement, beacon or gateway placement is also another important factor in WSN optimisation. The position in which gateways are deployed is crucial for connecting the overall network. Authors in [39] present an ILP model capable of locating the best beacon positions given a set of already selected sensor locations. Similarly, given a set of sensors, it is possible to determine the optimal gateway locations depending on the objective to optimize [40, 41]. However, the lack of clustering and sensor differentiation makes the model incomplete for IoT. In [42], a gateway placement with QoS constraints is presented in which clusters are created around the deployed gateways, which differs from our definition of cluster. In [43], authors show a model capable of selecting, from a given Wireless Mesh Network, the nodes that should act as gateways, instead of placing additional nodes.

2.3.2 Sensor Criticality

The problem of optimising WSN in multi-hop networks has been widely addressed in many research articles. Among them, different network metrics such as latency and lifetime are usually used as the main parameters for optimisation criteria. Moreover, Mixed Integer

Programming models cover the majority of the evaluations thanks to its precision when delivering results.

For instance, in [44], critical node removal impact in network latency is analysed by means of an MIP model inserted into a framework that iteratively removes critical nodes one after the other. Latency is calculated as the overall sum of the number of hops needed to transmit data from each node up to the base station. Since the problem is formulated and solved using a MIP model, it is time-consuming to calculate simultaneous removals. Due to this, authors only calculate the impact of the first and second most critical nodes in randomly generated networks with up to 100 sensors distributed in a disk shaped area of $200m$ radius.

Equally, research in [45] analyses the same impact by using similar input data and system model layout but focusing on network lifetime this time. Even though the MIP models are very similar, authors here go beyond the threshold of [44] by calculating the impact of up to five critical nodes in networks of 50 sensors with two different removal techniques: iteratively and simultaneously. In this case, lifetime is defined as the number of time slots needed until the first node runs out of battery life, taking into account that every node transmits one packet each time slot. Similarly, authors in [46] deeply study the case of eliminating the node that offers the most network degradation regarding latency. Results show the evolution of network degradation as the radius of the network decreases as well as when the number of nodes varies.

Another enhancement of the cited model is found in [47], where authors merge both metrics, latency and lifetime, and evaluate the network degradation in lifetime when limiting the overall number of hops, i.e., when limiting maximum latency. Studied network sizes vary from 30 to up to 90 sensors.

Even though MIP models are very good at delivering exact results for many optimisation criteria, they lack scalability. All the work cited beforehand share the same WSN structure: few network nodes and, overall, only the most and the two most critical nodes are studied.

In order to extend such scenarios, heuristics offer a good opportunity for obtaining close to optimal results in larger situations both in terms of network density and number of critical nodes. In this thesis, we try to cover this literature space that is yet to offer relevant results by presenting a GRASP meta-heuristic for networks up to 1000 nodes in a 630 radius area. These values are not arbitrary but scaled up from a base scenario utilised in [44] and [47] for calculating, respectively, the critical node impact in latency and lifetime.

2.4 Building Energy Management Systems

Building Energy Management Systems are gaining popularity as a crucial software tool for enhancing and optimising current buildings. The scenarios IoT systems enable and the possibility to control every element inside a building makes it possible to create centralised systems capable of taking intelligent decisions over not-so-smart devices.

Particularly, many efforts are being put in the research of smart decision-makers over HVACs and lights. Authors in [48] present a theoretical fuzzy-logic controller over HVACs to reduce energy consumption. A real HVAC control system is presented in [49] by means of a centralised policy scheduler that regulates indoor temperature with respect to current environmental parameters. However, manual user feedback is needed in order to adapt current scenario to occupant desires. In [50], authors state the opportunity of building a BEMS capable of aggregating several systems, instead of only controlling a single metric.

All the aforementioned literature falls short at delivering a BEMS that combines optimisation models with building occupancy and occupant actions. People inside the building represent a crucial metric for delivering, not only efficient energy management, but also acceptable indoor comfort levels with respect to occupants' desires.

For this, BEMS must take into consideration both metrics, and study the existing trade-off between them. It is important to detect such trade-off in order to adjust the BEMS to behave properly depending on the current state of the building and its occupants. Typically, energy consumed has been utilised as the main optimisation criteria. However, with the introduction of comfort, many authors have tried to come up with a reliable formula for calculating such occupant comfort. Due to the difficulty to integrate several independent comforts into a single formula, many authors have decided to aggregate such independent comforts into a weighted equation. For instance, authors in [51] propose the usage of a comfort formula combining thermal, luminosity, air quality and acoustics comforts.

In the specific case of the temperature, the comfort is usually calculated following the ANSI/ASHRAE Standard 55 [52] Predicted Mean Vote (PMV) equation, that can also be extended to introduce air quality aspects [53, 54, 55]. Recommended light levels of the National Optical Astronomy Observatory are shown in [56]; such recommendations have been utilised in many works as the comfort values to aim for [57, 58].

Chapter 3

Architecture Evolution

Preface

This chapter presents an historical evolution of IoT architectures defined since the first implementations that appeared. Initial architectures were mainly focused in local scenarios, with no global communication over the Internet. As the time passed, the architectures evolved into more global and standard ones. Currently, the vast proliferation of gadgets and the necessity to interconnect a vast amount of sensors makes it clear that the Cloud is the standard de facto to follow. Following this trend, this chapter defines and specifies a Cloud-based architecture used as the basis for our study, which tries to cover and revert the flaws of initial presented models. Firstly, a generic definition of the architecture is shown, and, then, the actual implementation used throughout the thesis is detailed.

3.1 Evolution of IoT Architectures

The rapid proliferation of gadgets, wearables and solutions for every problem present in our daily lives creates chaos inside the IoT. Companies are foreseeing all the market opportunities and they are delivering fast solutions by means of vertical architectures that disable the possibility to interconnect heterogeneous devices [59]. Moreover, the disagreement and lack of standardisation in communication protocols is making it impossible to create heterogeneous systems in which devices from different manufacturers interchange data smoothly without the necessity to create additional software elements.

This is known as the *vertical silos* problem: the creation of private and close solutions by every manufacturer, ranging from the physical layer up to the end user application. As shown in Figure 3.1, every manufacturer creates its own perception or sensing layer, which is responsible for data gathering. Then, data is managed by an own intermediate layer to, finally, deliver such data to the end user application. Such verticality does not allow the creation of an IoT solution as a combination of gadgets from different companies, to solve a more complex scenario. Following sections present an overview of the evolution of IoT architectures since the appearance of such term up until the current ones, making emphasis on how they coped with the stated *vertical silos* problem.

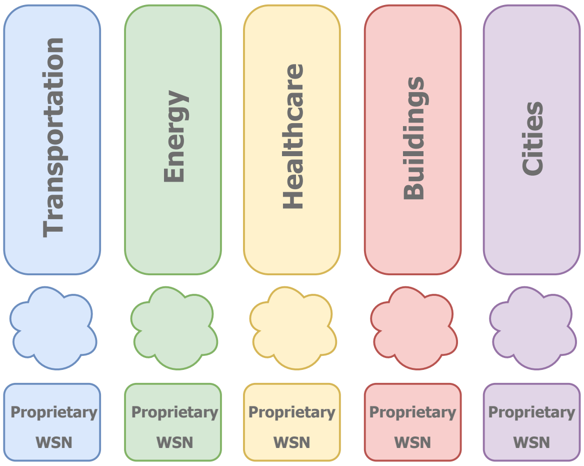


FIGURE 3.1: Abstraction of the vertical silos problem present in early stages of IoT development.

3.1.1 Initial models

The lack of architectural standards and protocols during the initial stages of the IoT development hampered the creation of systems with the current minimum requirements such as scalability, interoperability, security and reliability. During the first years, Intranet of Things was a more accurate term to define the situation. Devices were only provided with physical wireless communication protocols such as Bluetooth or ZigBee, with no possibility to transmit through the Internet. Moreover, the connection between those devices and the application was directly performed without any intermediate layer to decouple the system. Figure 3.2 shows an abstraction of such architecture.

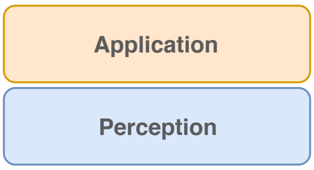


FIGURE 3.2: Initial IoT 2-Layer architecture.

As it can be seen, this architecture can only be divided into two separate layers, namely perception and application layer. Although it was possible to use multiple devices inside the same system, they acted as individual elements, without communicating between them or helping each other during transmission. That is, a network layer combining them into a WSN was not used. Instead, data being generated by the perception layer was directly sent to the application layer without any intermediate decoupling which made impossible the scalability or interoperability of such system.

Then, the architecture that can be seen as the birth of the Internet of Things appeared [60], which was comprised by three layers namely perception, network and application layer [61, 62, 63], as shown in Figure 3.3. The network layer grouped all the sensors and actuators of the system forming a WSN, in which devices were aware of each other. Additionally, gateways were added to gather and forward all the raw messages generated by the perception layer devices [49]. Even though initial systems continued only using physical wireless communication protocols for its devices, the insertion of gateways as a more powerful intermediate element allowed for Internet communication between the network and the application layers [64].

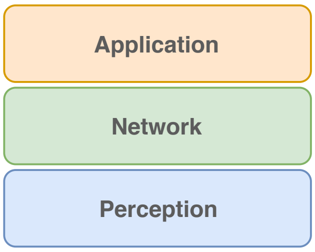


FIGURE 3.3: IoT 3-Layer architecture.

The introduction of the network layer helped to slightly cope with the scalability problem by means of the placement of more gateways, if necessary, to handle all the device connections. However, this did not completely solve the problem. Regarding interoperability and heterogeneity, the lack of standardisation between companies for the usage of protocols and message structure hampered the combination of several devices into the same system.

At this point, researchers and manufacturers agreed in the necessity of having an abstraction layer to completely decouple the physical network from the application, in order to allow the creation of device-agnostic applications.

3.1.2 The appearance of a *Middleware*

Many efforts have been put to provide an abstraction and standardisation layer from the WSN perspective. European Union projects such as SENSEI [65] and Internet of Things - Architecture (IoT-A) [66] have been addressing this problem by means of creating and defining the architecture for different applications. However, there is still a lack of agreement when it comes to overall architectural standards in regard to upper layers.

A middleware generally abstracts the complexities of the system and hardware allowing the application developer to fully focus all his efforts on the task to be solved without the distraction of concerns regarding system or hardware level [67, 68]. A middleware provides a software layer between physical and application layers [69, 70, 71, 72]. As it has been seen before, IoT interacts with many infrastructure and application technologies. Therefore, a middleware must provide almost full compatibility, among other characteristics [73, 74].

Even though of the agreement in the necessity of a middleware as an abstraction layer, during the last years, diverse solutions have appeared in terms of their design approach, such as event-based, database-oriented, application-specific or service-oriented [75, 76]. However, the usage of a single design approach might not be sufficient. Instead, successful middlewares have been built upon the combination of multiple designs.

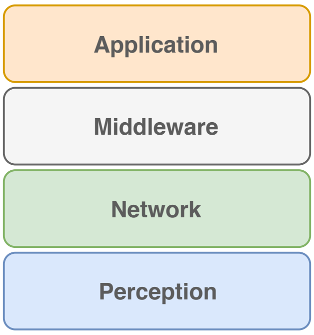


FIGURE 3.4: IoT 4-Layer architecture.

Since all the generated data must traverse the middleware for abstraction, the inclusion of the database as an element of such layer seems the right decision. This is why many current middleware solutions include a database-oriented design. However, the connectivity to the database may vary depending whether data is directly exposed to the end users or it is privately stored and, instead, events or services are offered. The former case follows a mere database-oriented design, whilst the latter is a combination of database and either event or service-oriented design.

Middlewares based on events with database storage are gaining popularity due to the easiness of deployment and lightness of resource utilisation. Since individual sensor messages can be seen as events, the storage is straightforward. Regarding event communication, this type of middlewares usually use the publish/subscribe pattern, in which a set of subscribers acquire events from a set of publishers. Protocols such as CoAP (Figure 2.1) or MQTT (Figure 2.2) are designed to this aim.

Service-oriented middlewares [70] are based on Service-Oriented Architectures [77] that has been traditionally used in corporate IT systems. Characteristics such as service reusability, composability or discoverability are also beneficial for IoT scenarios. However, large scale networks, constrained devices and mobility make this approach challenging.

With these approaches, applications connected to the middleware benefit from the abstraction and are agnostic to the underlying hardware.

3.1.3 Towards intelligent IoT systems

Up to this point, IoT applications started to exploit the benefits of new and well designed architectures to solve daily problems or make lives easier. Moreover, many monitoring applications for different scenarios appeared, such as health monitoring systems, building energy monitoring or city resource monitoring. However, the essence of such systems was merely informative.

New IoT or Future Internet is meant to go beyond that informative perspective [78]. Instead, creation of intelligent and autonomous systems is what companies and researchers are aiming for [79]. To this aim, new elements are added to the previous defined architecture.

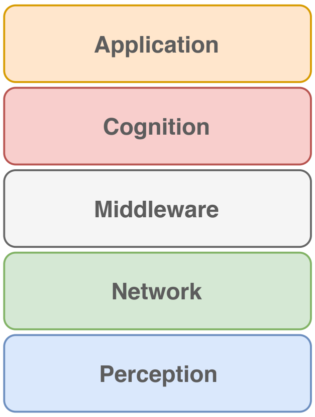


FIGURE 3.5: IoT 5-Layer architecture.

Specifically, a new layer appears between the middleware and the application, commonly named knowledge-based layer, context awareness layer or cognitive layer. It is responsible for requesting data and extracting valid information for acquiring new knowledge and act upon it [72].

Depending on the purpose of the application, many techniques can be used, such as rule based programming, machine learning [78] or predictive analysis. Rule-based applications are meant to modify the status of the scenario if certain events occur. Usually, rules are static. However, the combination of rules with machine learning techniques offers a richer system in which rules are modified depending on past actions. Predictive analysis is also being used to anticipate future actions and, for instance, increase building occupant comfort by adjusting indoor elements to the desirable state prior to the entrance of such occupants.

3.2 Cloud-based IoT Architecture

This section presents an architecture developed to try to cope with some of the flaws stated in the previous sections. It offers interoperability in regard to the type of sensors and protocols that can be used. Moreover, reliability and data persistence is achieved by means of a Cloud middleware capable of replicating services on demand. The Cloud also allows for data exposure and possibly, utilise it as a service for third parties.

Figure 3.6 shows the architecture broken down into different layers. Starting from the bottom, the perception layer includes all the sensors and actuators of the network. It is responsible for sensing the environment and also for executing the necessary actions that are received from the above layer.

Network layer comprises and groups the gateways of the platform. Since this devices are resource constrained in regard to the number of established connections, it is necessary to study the scenario under development in order to know how many of these devices need to be deployed. In terms of energy usage, gateways need a more powerful source than sensors.

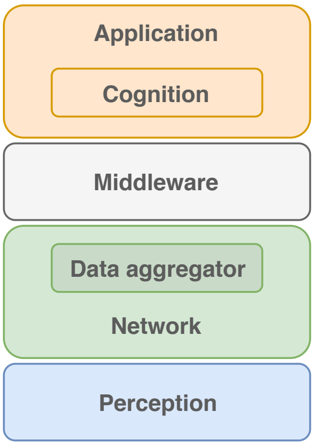


FIGURE 3.6: Cloud-based IoT architecture abstraction.

That is why these devices are usually placed inside buildings to maintain them fully operable. Moreover, received data might need to be uploaded to the Internet, which is another reason to locate them inside Internet-reachable buildings.

As previously mentioned, one of the main issues regarding IoT and WSN is the heterogeneity at the physical level. The lack of agreement for communication and message structure make it necessary to endow the system with the possibility to upgrade gateway software to allow compatibility with new devices.

Even though the south gate (i.e. the communication between sensors and gateways) of the gateways may be heterogeneous, the north gate (i.e. the communication between gateways and the above layer) maintains its homogeneity by exclusively using one communication protocol.

The data aggregator and processing layer, as its name says, is responsible for receiving every sensing message in raw format. These messages are then processed in order to modify their structure to a standard one. Since JSON is the standard de facto inside the Big Data world, and many non relational databases utilise it, raw sensing data is transformed into JSON formatted files. This layer can be seen as a module of the network layer, and that is why it has been located inside it.

The middleware plays an important role inside the architecture. As previously stated, the Cloud is the standard the facto to host such middlewares in order to offer worldwide connectivity and data delivery to any third party client. Middlewares must allow for data upload, storage and retrieval by means of standard protocols such as HTTP or MQTT. With the addition of big data technologies, they also offer the possibility for server and database replication in case of necessity due to an increase in the number of connection requests.

Finally, the application layer contains the actual application. It is fed with standard data incoming from the middleware which allows the developer not to worry about the underlying hardware and communication protocols. Note that the cognition layer is inserted into the application layer, in order to avoid coupling the rest of the architecture with specific smart logic.

This architecture has been used for the development of a simulator for smart buildings that tries to reduce building energy consumption by avoiding unnecessary and wasteful device states, while maintaining acceptable levels of occupant comfort. For instance, switching *off* room lights when it is empty or adjusting room temperature depending on both environmental conditions and occupant desires. It is worth mentioning that such architecture has been developed having in mind the interoperability, shareability and reutilisation of all the layers with the exception of the application one. That is, sensor data gathered and stored in the Cloud can be utilised by any third-party application that demands its usage, after the corresponding authentication for data retrieval.

Following sections explain more in detail the development of every layer and the communication between them.

3.2.1 Perception layer

The perception layer is formed by all the sensors and actuators of the system. The main task of this layer is gathering data from the elements of the scenario under monitoring. In the case of a Smart Building, sensors are usually placed to monitor environmental conditions such as temperature, humidity, luminosity, air quality, and also device states such as doors, windows, blinds, computers, etc. Moreover, actuators are deployed to allow status modification of those elements. For instance, if the system detects that a room has been left with lights *on*, it can send the signal to switch them *off* in order to avoid wasting energy unnecessarily. To achieve this feature, the communication between devices and the layer from above is bi-directional.

The system can also take advantage of the communication bi-directionality to interact with the sensors. Since sensors can sometimes be located in hard-reachable locations, this feature is needed to allow for software upgrade without having to manually access to them, commonly known as Over The Air (OTA) programming.

As it can be seen, there is a plethora of characteristics to monitor and actuate with, allowing for wide market opportunities for companies. The *vertical silos* problem previously stated starts in this layer. Companies usually specialise themselves in a single scenario or problem, without commonly agreeing standards in design or communication. However, when a more general system is developed such as a Smart Building, it is necessary to combine multiple sensors to fulfill all the requirements stated above. Therefore, it is needed a layer in which all this differences are solved by allowing the transmission and communication of multiple protocols.

3.2.2 Network layer

The network layer groups and manages the gateways and it is responsible for creating a WSN between sensors, actuators and gateways. Due to the heterogeneity of the perception layer, gateways must be rich in terms of protocol compatibility. To this aim, they must be endowed with multiple interfaces depending on the type of sensors they are managing. Due to this necessity, their requirement in terms of energy usage is higher and the usage of batteries is not sufficient. Instead, they are deployed inside buildings in order to be connected to the electricity grid.

Up to this point, the communication between the devices of the system is locally performed. However, once messages are received by the gateway, the connectivity can vary. Local systems can opt to maintain a private network with no Internet connection in which gateways locally connect to the above layer to standardise and store messages. Another approach could be to endow the gateways with Wi-Fi interfaces for direct data upload to the Internet.

3.2.3 Data aggregator layer

The data aggregator layer can be seen as the standardisation message layer. It is responsible for receiving raw messages from every gateway and transforming them into a standard message format. It has been decided to use JSON as the data standard because of the compatibility with the above layer and the friendliness that it offers with big data technologies and no relational databases.

This layer can be deployed into multiple places inside the architecture. Specifically, these are valid locations for it:

Distributed Multiple instances distributed across the gateways.

Centralised Central server with replicability for connection handling.

Middleware Separated module inside the middleware.

Depending on the power of the gateways, this layer can be deployed in each of them in order to avoid the necessity of a central server gathering the data from every gateway to later transform and upload it. However, this decision has some drawbacks. Firstly, it requires that all the gateways of the platform are capable of connecting to the Internet in order to upload the data. Secondly, processing power and storage for these gateways would need to be higher. To conclude, in the case of needing to make a modification in the data aggregator to allow new data structures, it would be needed to completely flash all the gateways of the platform.

Another alternative is to develop a module for the middleware under usage in order to have a unique layer capable of standardising the data and storing it. This is a good design approach but in our case it has not been followed in order to maintain the external middleware intact.

The design approach finally followed has been to deploy this layer into a central server capable of creating multiple replicas if needed to cope with incoming connections. With this design, gateways do not need to have Internet connection and their processing power can be reduced to also optimise energy consumption. Also, this allows to deploy gateways powered solely by batteries in the event of strict necessity.

3.2.4 Middleware

As previously defined in Section 3.1.2, a middleware is an abstraction layer that hides the complexities of the system and hardware underneath. That is, applications and end user software do not need to know how the data is generated or modified throughout its way up to the middleware.

The following list presents a set of features required for the specification of a Cloud middleware:

Cloud Storage is carried by the Cloud with standard technologies.

Big Data Oriented to the usage of Big Data technologies with high scalability.

Standard Usage of standard communication protocols for data upload and download.

Security Public and private virtual objects for data scope control and sharing.

An initial and basic feature is the utilisation of standard technologies. As previously stated, this middleware acts as a hiding layer to avoid the propagation of sensor and communication protocol heterogeneity. For this, the middleware must utilise standard technologies for storage, such as *key: value* JSON files, and standard communication protocols for offering data, such as REST API and publish/subscribe protocols.

Moreover, since this middleware can be utilised anywhere thanks to the Internet, it is mandatory to deliver big data solutions with high scalability, to guarantee that every connected client receives the information as fast as possible. Note that many IoT systems cannot allow the loss of any message due to their criticality.

Security and privacy are important concerns present in every recent application. Due to the exposure of sensor data to the Internet, it is important to guarantee and offer security levels for the stored messages in order to avoid unwanted clients to access sensitive data. In particular, two main scopes are considered, namely public and private. Public scope allows for the access to the data by any third-party application without the need of previous authentication or identification. On the contrary, private scope forces the authentication of the application in order to send sensitive information. To implement such behaviour, messages stored in the Cloud middleware contain a *key*. This key is granted when the first connection is established. Then, any third-party application must include this key inside its subscription petition in order to receive message updates from the *key-protected* sensors.

3.2.5 Application layer

The application layer, as its name indicates, contains the application responsible for interacting with the user or showing the desired information. Inside the IoT world, current developed applications are firstly focused on monitoring the environment and acting as an information panel in which the user can read, in real time, the values of the different sensors of the system, such as the indoor temperature, power usage, outdoor luminosity, etc. There also exist interactive applications in which the user, apart from being able to see the sensor information, can also interact with the environment by performing actions such as close the door, lower the inside temperature or switch elements *on* or *off*.

One of the main advantages of the architecture presented is the freedom that the developer has when creating a specific application. By having a middleware with standard formats and transmission protocols, it allows him to fully focus their efforts into the use case without having to take into consideration hardware specifications.

The only coupling element between the middleware and the application is the message reception module. As it has been previously mentioned, messages can be requested via REST API or subscriptions thanks to the publish/subscribe protocol. The former allows for synchronous data requests, which can be necessary when a specific value needs to be obtained. However, the later is the standard de facto used in modern IoT applications. The publish/subscribe protocol allows for asynchronous message reception without the need to constantly query the middleware. Instead, when a new sensor message is stored inside the middleware, it is directly forwarded to the application by means of the subscription previously performed.

3.3 Implementation

This section converts the general guidelines presented in Section 3.2 into a more detailed implementation of the Cloud-based architecture. The aim is to explain more in depth the developed Cloud-based architecture that can be then utilised in any scenario. In our case, the implementation is oriented towards a Building Energy Management System. Figure 3.7 depicts all the elements that comprises the overall architecture framework.

Due to the high cost of deploying a real scenario with all the required sensors, it has been decided to test such implementation in a reduced testbed in order to check the proper behaviour of the overall IoT architecture.

Starting from the bottom, the perception layer aggregates all the sensors responsible for gathering surrounding data. Specifically, three AdvanticSys XM1000 sensors [80] are deployed in distinct rooms of a university building. Sensors are responsible for gathering temperature, luminosity and humidity data. The aggregation of these sensors creates a Wireless Sensor Network, that can either be homogeneous or heterogeneous. It is important to detect the type of network needed for every scenario in order to grant the network with enhanced features to cover two of the most important WSN metrics, namely reliability and resilience. Networks must contain an acceptable level of protection and backup plans to avoid unexpected partial network disruptions. Depending on the type of network, the detection of potential critical WSN regions requires different approaches. Additionally, the placement of the sensors inside the area under study must also be accurately planned in order to cover the data gathering requirements the system might have. In the next chapters, a tool for solving such problem in both network types is presented. Specifically, Chapter 4 studies the sensor placement and

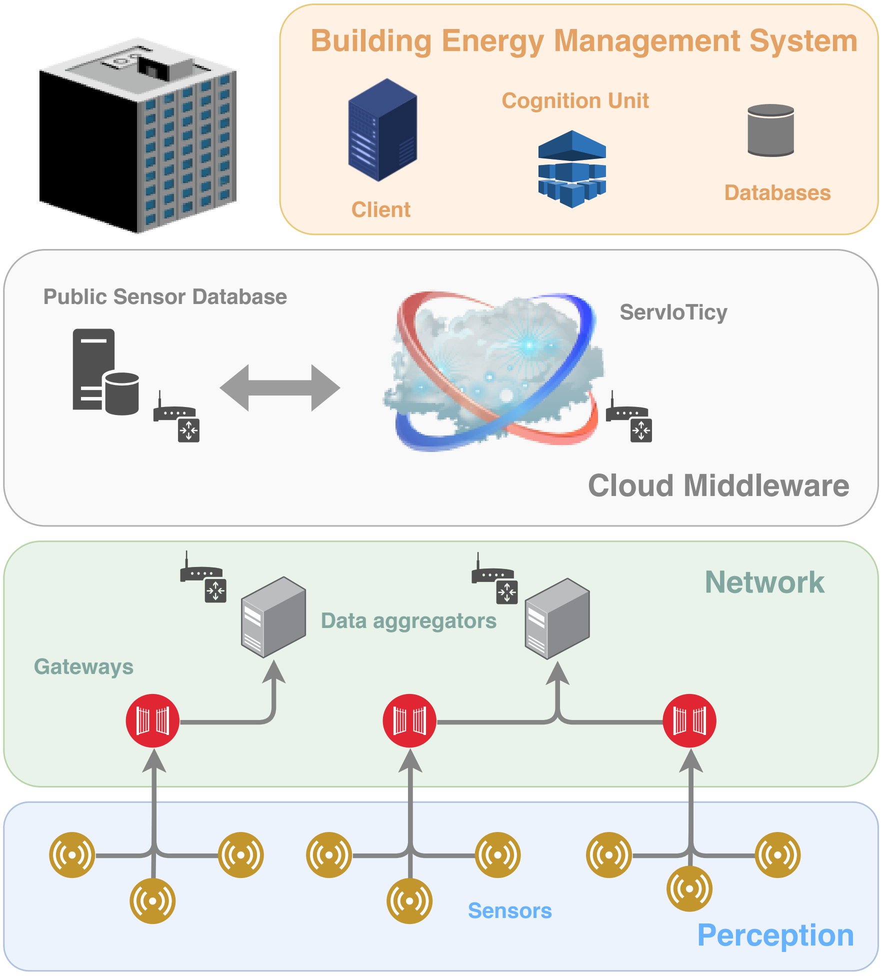


FIGURE 3.7: Cloud-based IoT architecture.

protection enhancement of heterogeneous WSN. Then, Chapter 5 considers the detection of critical sensors in a homogeneous WSN.

In the specific case of our testbed, IoT network can be referred to as an homogeneous WSN, since the utilised physical sensors are composed by several smaller sensors for gathering all the aforementioned environmental values. However, real IoT building systems require the placement of heterogeneous sensors in order to gather yet more metrics.

Data gathered by the sensors is directly sent to the network layer. In particular, it is sent to a centralised gateway connected to a server. Note that such gateway is physically identical to the rest of the sensors, but it has been flashed to only act as a data receiver. Additionally, such gateway can be nourished by server electric power in order to make its battery last longer. Sensing units are endowed with independent batteries. Sensor messages are delivered to the gateway by utilising a proprietary communication protocol and specific data structure.

Once all the data is received by the sensors acting as gateways, it is directly forwarded to the data aggregator tool deployed inside the server. Data arrives in raw format, and it must be processed in order to grant the Cloud middleware with normalised sensor messages in standard formats. In our specific case, data is transformed into *{key: value}* JSON formatted files.

The Cloud middelware is formed by two major software tools. Data storage is performed by a platform called ServIoTicy [81], which covers all the requirements stated in Section 3.2.4. ServIoTicy is an online platform developed by the Barcelona Supercomputer Center [82] during the COMPOSE project [83]. It allows for fast and simple composition of IoT data streams, offering multi-tenant data architecture. As for its communication capabilities, both for data upload and download, it allows REST and publish/subscribe communication. Figure 3.8 shows an abstraction of its behaviour and possibilities.

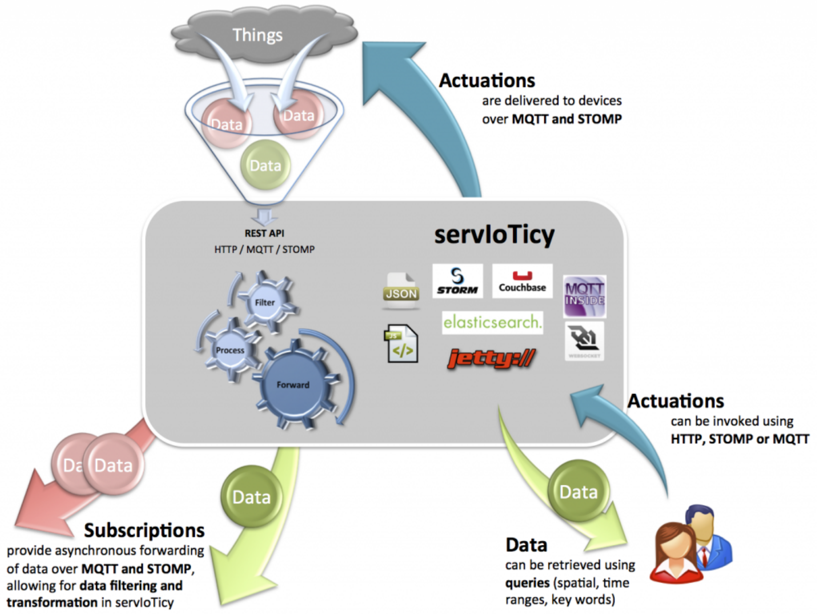


FIGURE 3.8: Abstraction of the ServIoTicy platform [84].

A sensor storing data in the Cloud middleware is modelled as a virtual object with a globally unique identifier. For this, the server is responsible for requesting a new unique sensor identifier each time data from a new sensor is received. In addition to that, each sensor must specify a visibility level. Currently, public and private visibility scopes are available. The former does not require any additional action by the server when requesting a new sensor identifier. The latter, however, requires the specification of private scope when requesting a new identifier to ServIoTicy. Whilst public scope allows any third-party application to subscribe to the public sensor and receive messages, private scope only allows it to the third-party application that utilises the same *API KEY* as the private sensor.

Now that the public scope is presented, it is possible to define the other important element of the Cloud middleware, namely public sensor database. The public sensor database is created to cope, as much as possible, with the aforementioned number of connected devices problem. It can be seen as a catalogue of sensors deployed under certain areas and publicly available.

Field	Description
serviotic_id	Public service object ID granted by ServIoTicy
model	Sensor brand and sensing capabilities
location	Physical location of the sensor

Table 3.1: Description of the public sensor database parameters.

Table 3.1 describes its parameters. Particularly, the *serviotic_id* permits the subscription to ServIoTicy in order to receive real-time sensor messages. The *model* offers information about the sensor brand and the data it collects. Finally, the *location* specifies the physical location of the sensor. Such location is currently defined by utilising the same university room names. However, once the IoT system is globally deployed, physical coordinates are the way to go, since they permit unique global identification.

Once the server requests and receives a new service object identifier, it is stored inside the public sensor database along with its hardware and location information. As can be seen, the Cloud middleware contains all the necessary information to query any public sensor without having to control any physical device. Thus, the middleware completely obfuscates any possible underlying heterogeneity.

As for the application layer, a Building Energy Management System is built in order to connect and receive updated sensor messages via the Cloud middleware. Specifically, the BEMS initially connects to the public sensor database for obtaining the subset of service object identifiers required for the correct functioning of the system. Then, the BEMS subscribes to such identifiers to receive real-time data. The BEMS is defined more in depth in Chapter 6. However, for this testbed, simulation capabilities are cut, and only message reception is performed. The utilisation of the same BEMS in the testbed as in the simulations shows that, in the event of deploying the system with many sensors, it is possible to adapt the current tool with little effort.

Chapter 4

Sensor Placement and Deployment

Preface

This chapter introduces and specifies the first tackled problem related to the planning phase of an IoT system. Particularly, the well known Wireless Sensor Network placement problem is studied and enhanced with new requirements arising from their mandatory use in any IoT system. Moreover, the importance of data in such new systems demands high levels of network availability and reliability. To this aim, protection constraints are also added to the definition of the problem in order to grant to the network potential backup solutions in the event of unexpected partial network disruptions.

4.1 Introduction

Wireless Sensor Networks have been in the spotlight for many years, but the appearance of IoT has increased their usefulness to yet unknown levels. The necessity to monitor every possible device inside a given scenario has increased the popularity of WSN. However, the requirements of such networks need to evolve in order to make them compatible with new IoT systems.

WSNs can be divided into two major groups. Homogeneous WSNs consist of the aggregation of equal sensors in terms of connectivity protocols and sensing characteristics. On the contrary, heterogeneous networks are made of sensors with distinct sensing capabilities and communication protocols.

Initially, IoT systems were made of homogeneous WSNs capable of sensing a specific metric under a certain environment to later extract conclusions and valuable information. For instance, pollution can be monitored in different city points to gather air quality and decide whether traffic needs to be controlled under certain areas.

On the contrary, complex IoT systems need the aggregation of multiple data from different sources. For instance, a smart factory must continuously monitor the status of its machines in order to detect anomalies. Moreover, environmental sensors for temperature, humidity and pollution are needed to detect any possible anomaly such as fire or lack of oxygen.

As it can be seen, both WSN types will be used in the future for building smart systems. However, their placement and deployment requirements vary. We identify such requirements and present, for each case, a solution for the problem under study. This chapter extends the problem of localising the optimal locations for sensor deployment in heterogeneous networks by introducing clustering requirements. Since no every sensor can be deployed anywhere, it is needed to restrict the types of sensors to be deployed in each possible location in order to be able to sense each desired metric in each desired scenario space for gathering all the data correctly. Additionally, such planning allows for the utilisation of the least possible number of sensors. In addition to that, a protection requirement is also introduced to enhance the WSN with backup capabilities in the event of partial disruptions. The impact of such protection is further analysed.

Homogeneous networks are studied and explained more in depth in Chapter 5.

4.2 Heterogeneous Placement Problem

This section presents and defines the placement problem for heterogeneous WSNs. As previously explained, we extend the problem of covering a sensing area with the minimum number of sensors by introducing clustering requirements. Due to the characteristics of complex IoT systems, each zone of the scenario under study is represented as a cluster, and each cluster is defined by the number and type of the sensors it must contain.

Section 4.2.1 contains a more formal definition of the problem. Furthermore, in Section 4.2.2, the problem is modelled using an Integer Linear Program with all the requirements explicitly stated. In addition to such model, a building layout is shown to clearly understand important aspects of the problem, such as the device positions and clustering representation.

4.2.1 Problem Statement

Given a building layout with known possible physical positions for placing either sensors or gateways, the problem resides in selecting the optimal locations for their deployment, minimising the energy consumption of the whole system while fulfilling connectivity, resource, protection, and clustering coverage constraints. As the transmission range of a sensor is limited to few metres, connectivity constraint ensures that a sensor is able to reach to at least one gateway. The resource constraint is referring to the maximum bandwidth of the gateway, i.e., the number of concurrent transmissions from sensors a gateway can receive. The protection constraint is related to unexpected gateway failures, meaning that if a gateway fails, the sensors initially transmitting to it need to be able to reach another gateway nearby. Finally, a cluster is defined as an area of the building where a given set of sensor types are mandatorily required. It is worth mentioning that the size of the clusters and the necessary sensors inside each of them varies depending on the zone under representation. For instance, just a single alarm is placed in each floor of the building, so then, in this case, the cluster represents a floor; temperature, presence, and luminosity sensors are placed in each room, which form room clusters; a single humidity sensor is installed each four rooms to create clusters of room groups, etc. Besides, the possible sensor positions are not exclusively attached to a single cluster; instead, they can be shared in order to create a cluster hierarchy, which specifies different cluster sizes that cover different building zones with distinct sensor types.

Given these constraints, the objective of the model is, therefore, to decide which positions to select in order to minimise the energy consumption determined as the power needed by the sensors to transmit to the gateway and the power required by the gateways to collect these data and retransmit it to the building manager.

4.2.2 Model Formulation

This section presents the WSN model that determines the sensor and gateway positions inside of a building aiming at minimising the overall energy consumed.

We represent a building scenario as a set of potential sensor positions $S = \{1, \ldots, N_s\}$ and a set of potential gateway positions $G = \{1, \ldots, N_g\}$, each of which corresponds to a 2D or 3D point depending on the necessity. Once the coordinates are marked, the variable *dist* of size $|S|x|G|$ defines, for each element $dist_{ij}$, the Euclidean distance between positions S_i and G_j. Figure 4.1 shows an example of a building layout with the available positions drawn, in which blue and red dots represent sensor and gateway positions, respectively.

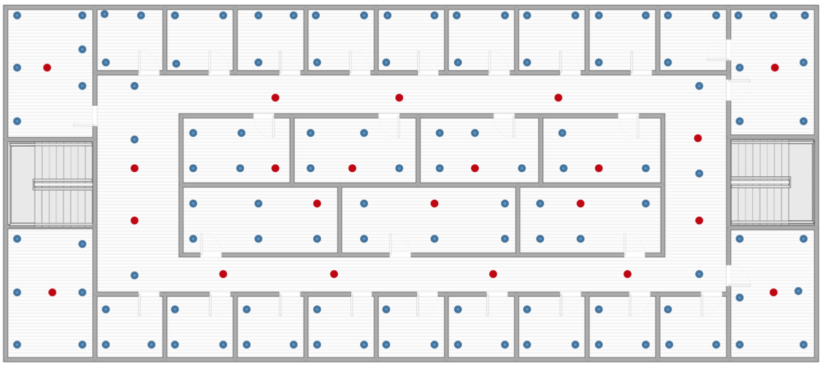


FIGURE 4.1: Layout of the building with the sensor and gateway available positions defined.

The determination of the locations to place sensors and gateway initially depends on their energy consumption model. The calculation of the sensors' energy consumption is directly related to the distance between sensors and their assigned gateway. We use Mica2 motes [85] due to their flexibility to alter their transmission range in order to efficiently manage energy consumption, as shown in Table 4.1. Specifically, $transmissionCost_i$ stores the energy needed for sending data using level i. To simplify further calculations, it is assumed that the energy of each level of the table corresponds to the transmission of a single data packet.

For the energy consumption of the gateway, we consider the parameter $CostG_j$. This cost is assigned to each gateway inside the range [400, 1000], utilising the same consumption units as the sensor transmissions.

l	$E_{tx}(l)$	$R(l)$	l	$E_{tx}(l)$	$R(l)$
1 (l_{min})	671.88	19.3	14	843.75	41.19
2	687.50	20.46	15	867.19	43.67
3	703.13	21.69	16	1078.13	46.29
4	705.73	22.69	17	1132.81	49.07
5	710.94	24.38	18	1135.42	52.01
6	723.96	25.84	19	1179.69	55.13
7	726.56	27.39	20	1234.38	58.44
8	742.19	29.03	21	1312.50	61.95
9	757.81	30.78	22	1343.75	65.67
10	773.44	32.62	23	1445.31	69.61
11	789.06	34.58	24	1500.01	73.79
12	812.50	36.66	25	1664.06	78.22
13	828.13	38.86	26 (l_{max})	1984.38	82.92

TABLE 4.1: Transmission energy consumption ($E_{tx}(l)$ in nJ/bit) and transmission range ($R_{(}l)$ in m) at each power level (l) for the Mica2 motes as a function of power level [85]. Energy dissipation for reception of data is constant ($E_{rx} = 922nJ/bit$).

Another important aspect to take into account is the variety of the sensors to deploy. IoT appeared thanks to the improvements made in sensors, which led to the possibility to gather any kind of data, such as temperature, humidity, luminosity, air quality, fire, etc. Our model handles T types of sensors, each of which is responsible for gathering different environmental data. Even though their characteristics may vary, we consider that the transmission range and energy consumption are equal and correspond to the aforementioned Table 4.1. For the transmission, the maximum possible reachable range is limited to $R(l_{max})$. To ease the final model, the adjacency matrix $InRange$ is defined:

$$InRange_{ijk} = \begin{cases} 1 & \text{if } i \in S, j \in G, k \in T, dist(i,j) \leq R(l_{max}) \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

This adjacency matrix allows for rapid knowledge of the possible connections that can be established between a sensor of type k placed in position S_i with a gateway placed in position G_j.

A novelty introduced in our WSN modelling is the utilisation of clusters. Clusters are usually defined as a group of the same or similar elements gathered closely. In our specific case, a cluster is defined as a set of sensor positions in which a group of sensor types needs to be exhaustively represented. This addition allows the creation of different solutions depending on the necessity, and the possibility to change the final layout inside the same building. The model considers N_c clusters, and the variable $clusterPositions$ stores, for each sensor position i, a list of cluster identifiers j to which such position belongs. This definition enables the possibility to create regions that belong to more than one cluster, in order to describe more realistic scenarios.

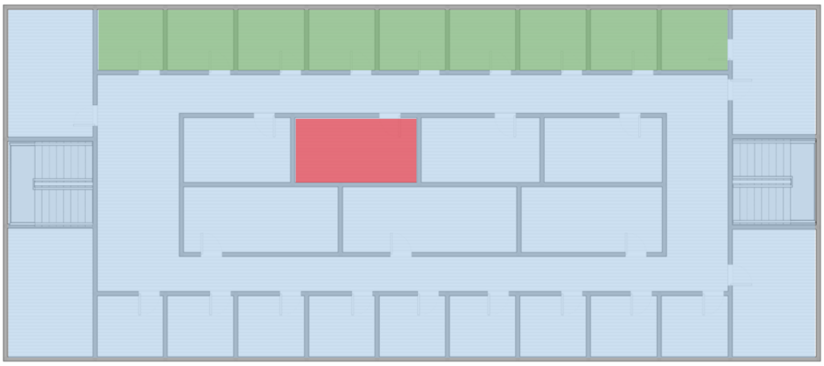


FIGURE 4.2: Example of three clusters of different size defined in a building floor.

Figure 4.2 shows an example of a cluster division. As it can be seen, the model permits the definition of small clusters inside bigger ones. In this particular example, the entire floor belongs to the blue cluster which allows the definition of sensor types that only need to be placed once per floor, such as a fire alarm. The green cluster consists of all rooms facing the top side of the building where sensors gathering information that do not significantly vary from one room to the contiguous ones, like air quality, humidity and gases detection are placed. The red cluster represents a single room, where sensors like temperature, presence, luminosity, smart plugs, etc. have to be installed.

Since the number of positions inside each cluster may vary, the final model uses an auxiliary adjacency matrix *clu*.

$$clu_{ij} = \begin{cases} 1 & \text{if } i \in S, 1 \le j \le N_c, j \in clusterPositions_i \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

As it can be seen from Equation 4.2, each clu_{ij} determines whether position S_i is inside the cluster with $id = j$.

In order to handle heterogeneous scenarios, another variable is needed to indicate the set of sensor types that each cluster must contain. To this aim, the variable $clusterTypes$ is defined, and stores, for each cluster i, a list of sensor types k that must be present. As before, since the number of sensor types needed per cluster may vary, an auxiliary adjacency matrix is used for the final model:

$$cluTypes_{ij} = \begin{cases} 1 & \text{if } 1 \le i \le N_c, j \in T, j \in clusterTypes_i \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

A final valid deployment is represented by a set of sensor locations $PS = \{1, \ldots, N_s\}$ and a set of gateway locations $PG = \{1, \ldots, N_g\}$. To this aim, the following set of boolean decision variables is defined to know the selected gateway and sensor positions with their corresponding type, as well as the link between them.

$$x_{ij} = \begin{cases} 1 & \text{if a sensor of type } j \text{ is installed at position } S_i \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$y_i = \begin{cases} 1 & \text{if a gateway is installed at position } G_i \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

$$a_{ijk} = \begin{cases} 1 & \text{if sensor } i \text{ of type } k \text{ is attached to gateway } j \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

$$b_{ijk} = \begin{cases} 1 & \text{if sensor } i \text{ of type } k \text{ is attached to gateway } j \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Equation 4.4 shows the boolean variable that indicates whether a position S_i contains a sensor of type j. Similarly, Equation 4.5 defines the analogous variable for the gateways. However, since there only exist one type of gateway, only one dimension is needed. Last variables are presented in Equations 4.6 and 4.7, and indicate the wireless link connections between sensors and gateways. Specifically, it determines whether a sensor in position S_i of type k is attached to a gateway positioned in G_j for the active transmissions and the backup transmissions, respectively.

With all the decision variables defined, it is possible to clearly state the cost of deploying a sensor in terms of its associated gateway and the distance between them. To do so, we firstly calculate the *index* used for acquiring the proper transmission level as shown in Table 4.1. For sake of simplicity, we assume that distances are rounded to the next unit.

$$index = \lceil a_{ijk} * dist(i,j) \rceil \tag{4.8}$$

As it can be seen in Equation 4.8, the index calculation is controlled by the decision variable a_{ijk}, in order to only count the transmissions in which sensor i of type k is attached to gateway j. It is worth mentioning that the index is always inside the range $[0, \lceil R(l_{max}) \rceil]$. For $index = 0$, Table 4.1 is not defined. However, this value is only obtained when $a_{ijk} = 0$. Thus, it is safe to define $transmissionCost[0] = 0$. Once the index is defined, the $transmissionCost[index]$ returns the energy consumed for such transmission.

With all the required parameters and the variables defined, the proposed Mixed Integer Programming (MIP) model for solving the problem is as follows:

$$\min \sum_{i \in S} \sum_{k \in T} x_{ik} * \sum_{j \in G} transmissionCost[index] + \sum_{j \in G} CostG_j * y_j \tag{4.9}$$

subject to

$$2 * a_{ijk} \leq x_{ik} + y_j \;\; \forall i \in S,\, j \in G,\, k \in T \tag{4.10}$$

$$2 * b_{ijk} \leq x_{ik} + y_j \;\; \forall i \in S,\, j \in G,\, k \in T \tag{4.11}$$

$$\sum_{j \in T} x_{ij} \leq 1 \; \forall i \in S \tag{4.12}$$

$$a_{ijk} \leq InRange_{ijk} \;\; \forall i \in S,\, j \in G,\, k \in T \tag{4.13}$$

$$b_{ijk} \leq InRange_{ijk} \;\; \forall i \in S,\, j \in G,\, k \in T \tag{4.14}$$

$$\sum_{j \in G} a_{ijk} \geq x_{ik} \;\; \forall i \in S,\, k \in T \tag{4.15}$$

$$\sum_{j \in G} b_{ijk} \geq x_{ik} * (protectionLevel - 1) \;\; \forall i \in S,\, k \in T \tag{4.16}$$

$$a_{ijk} + b_{ijk} \leq 1 \;\; \forall i \in S,\, j \in G,\, k \in T \tag{4.17}$$

$$\sum_{i \in S} \sum_{k \in T} a_{ijk} + b_{ijk} \leq \;\; GatewayBw \;\; \forall j \in G \tag{4.18}$$

$$\sum_{k \in S} clu_{ki} * x_{kj} \geq cluTypes_{ij} \;\; \forall i \in C,\, j \in T \tag{4.19}$$

Objective function, defined in Equation 4.9, states the minimisation of the whole installation cost by adding the consumed energy for every sensor-to-gateway single packet transmission, as previously indicated, and the cost of deploying each gateway.

Equations 4.10 and 4.11 ensure that, if sensor position S_i is associated with gateway location G_j, then, a sensor must exist at position S_i and a gateway must be placed at position G_j for both the active connections and the backup ones, respectively.

Equation 4.12 ensures that each position S_i holds one and only one type of sensor, since the same physical position cannot be used by two separate elements. Note that this constraint refers to single physical elements, but a type of sensor can be defined as an aggregation of multiple single sensing units.

Equations 4.13 and 4.14 check that all sensor-to-gateway links created are in the transmission range of the sensor, for both the active connection and the backup ones.

Protection level is defined to introduce resilience to the network in case of necessity. This level indicates the number of gateways to which each sensor must be in range of, in such a way that, if a gateway fails, the sensor can be connected to another one. To do so, Equations 4.15 and 4.16 force a sensor of type k at position S_i, to be connected to at least *protectionLevel* gateways. Specifically, *protection level - 1* different gateways as backup connections, and 1 gateway as the active target. Additionally, Equation 4.17 is responsible for ensuring that no connections between sensors and gateways are counted both as an active connection and a backup connection. Therefore, it guarantees that *protection level* different gateways are attached to each sensor. It is worth mentioning that the transmission level between such available connections can vary.

Bandwidth constraint, defined in Equation 4.18, ensures that all gateways handle less than their maximum bandwidth. For sake of simplicity, we define the maximum bandwidth of a gateway as the maximum number of sensors connected to it. In case a given protection level is applied, the backup connections are also considered since they need to be available in the event of a gateway failure.

Finally, clustering constraint Equation 4.19 forces each cluster to contain all sensor types required by the problem definition.

Note that the distinction between the active sensor-to-gateway connection and the backup connections allows for only taking into consideration the energy consumed by the active connections in order to obtain the optimal result, since such connections are meant to be the most used ones. The utilisation of backup connections is only activated in the event of a network failure, and the time span of the failure should be short with respect to the time span of the normal network behaviour, until the failure is fixed. Due to this, energy consumed by backup connections is not considered for the optimisation.

4.3 Deployment evaluation

This section presents an evaluation of the WSN deployment problem previously defined. It specifically aims at demonstrating that the proposed planning model effectively provides energy savings while fulfilling all the required constraints. Due to the lack of ability to extract conclusions only from the proposed solution, a comparison against a hypothetical complete deployment where all types of sensors are placed, when possible, in each building room, is shown. Such comparison is presented in terms of number of sensors and gateways deployed as well as overall transmission energy.

The comparison against a complete deployment makes sense due to the inability to conclude whether a sensor is necessary or not in a hypothetical manual installation.

After running the software with all the possible deployment locations defined in Figure 4.1, the optimisation model selects the positions shown in Figure 4.3 for the deployment of sensors and gateways. Green positions indicate sensors whilst orange ones are gateways. This figure considers that there is no protection level. That means that the protection level is 1, and, thus, each sensor is only guaranteed to reach one gateway.

Table 4.2 shows the number of sensors and gateways deployed as well as the corresponding energy consumption for both our proposed solution and the complete configuration taken as reference, where all the building rooms are covered with all the required sensors, if possible, and gateways are installed in all their possible locations. For the sake of comparison simplicity, it is considered that all the sensors transmit the same amount of data during the same amount of time. Also, device consumption is not taken into consideration since the number of required devices varies from one solution to another.

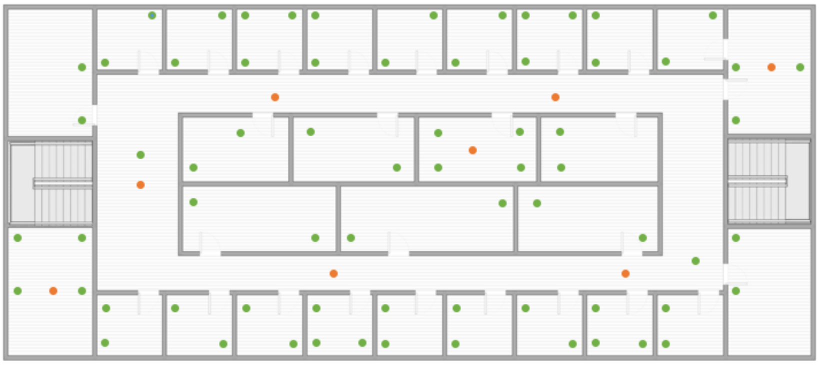


FIGURE 4.3: Sensor and gateway locations chosen by the model for the final optimised deployment.

As it can be seen, by accurately defining system requirements, it is possible to optimise the placement of sensors while still fulfilling them. Particularly, it is possible to reduce the number of installed sensors and gateways by 36% and 63%, respectively. This reduction is also visible in the energy consumed, in which it is possible to achieve a 38% energy saving for this specific building evaluation.

Network	Sensors	Gateways	Transmission Energy (nJ)
Optimised	69	8	54,076
Complete	108	22	86,154

TABLE 4.2: Number of sensors and gateways installed and transmission energy consumption, according to values of Table 4.1, for the optimised and the complete network result.

Even though the proposed solution presents better results in terms of deployed devices and energy consumed, it is also worth studying network failures.

In the event of a partial network failure, it can be inferred that the optimised solution might not be capable to deliver the complete network data. If a sensor is shut down, the data from its monitored devices is lost due to the lack of backup sensors. In the case of a gateway failure, it is possible to disconnect part of the network from the sink node and lose all the data collected by the disconnected sensors. The complete deployment, however, is more robust against network failures due to the amount of different transmission paths each sensor has to the sink node enabled by the amount of installed gateways.

To achieve similar network resilience, different levels of protection are studied. Remind that the protection level defines the number of gateways each sensor must be able to reach. For instance, if the protection level equals 1, sensors can only reach a single gateway; at protection level 3, three gateways are reachable and only the simultaneous failure of all three would disrupt the communication. This increases network robustness in the event of a gateway failure, since sensors can redirect their data to the closest available gateway. Note that the higher the protection level, the higher the network resilience against unexpected failures.

Protection level	1	2	3	4	5	6
Complete	86,154	86,154	86,154	86,154	86,154	86,154
Optimised	54,076	54,263	54,635	55,632	58,310	60,852
Failure	54,076	62,504	65,300	66,401	69,676	70,411

TABLE 4.3: Energy consumed in nJ for the different network configurations and protection levels.

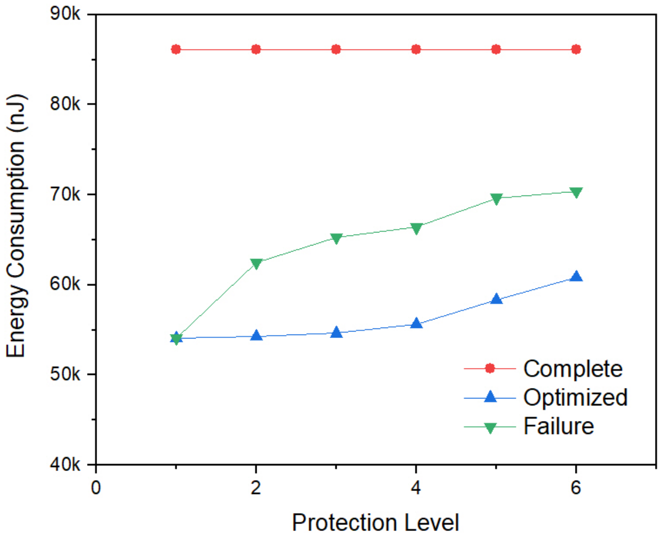


FIGURE 4.4: Network transmission energy consumption for the optimal and failure curves for different protection levels.

Table 4.3 and its graphical representation in Figure 4.4 show the effect of increasing the protection level of the WSN. Specifically, three different cases are depicted. The *optimised* and *complete* curves make reference to the aforementioned proposed models, being the first the one with no protection level, and the second the one with all sensors and gateways deployed, if possible. Note that the energy consumption for the *complete* configuration is independent of the protection level since the deployed network is constant throughout all levels.

In addition, a *failure* curve is shown, which considers the cases where *protection level - 1* gateways fail simultaneously and the sensors must, therefore, transmit the data to the closest available gateway. The removal of *protection level - 1* gateways is considered in order to ensure that the network remains connected. In the event of removing *protection level* gateways, there might be a sensor having all the removed gateways as backups, and, thus, such sensor is completely disconnected from the network, with no available backup paths. Additionally, for each protection level, the worst case scenario is considered. That is, the set of *protection level - 1* gateways removed from the network has the highest incidence in the overall energy consumption. As it can be observed, the *optimised* and *failure* curves grow as the protection level increases, due to the increment induced by gateway failures on the sensor-to-gateway transmission distance. Even though the *failure* curve always presents more overall energy consumption compared to the *optimised* one, the difference is not constant when incrementing the protection level. Indeed, when removing just one gateway, i.e., protection level equals 2, it can be seen that the increase of energy consumed is substantial with respect to the rest of protection levels. However, as the protection level increases, such increment is lower due to the path possibilities that new deployed backup gateways offer. Therefore, it is safe to say that the introduction of new gateways helps to reduce the increment in energy consumption when removing a set of them.

Moreover, it can be noticed that, if the resilience and availability requirements of the network demands the utilisation of a protection level, setting it to 3 is reasonable due to the small additional energy consumption induced by increasing it from 2 to 3. From there, the increment in energy consumption when increasing the protection level is almost linear. Then, the utilisation of a higher protection level depends on the level of resilience the network must guarantee. Protection levels beyond 6 have not been studied due to the inability of our model to obtain results. The amount of required backup connections and the gateway bandwidth restrictions make it infeasible to consider higher protection levels.

It is worth mentioning that the obtained results are specific to the network layout presented in Figure 4.1, with transmission energy consumptions shown in Table 4.1 and predefined gateway deployment cost. For another input parameters, the behaviour of the protection level may vary. From the obtained results, it is clear that there exists a trade-off between transmission energy consumption and protection level. Since IoT WSNs may be used for many applications as previously stated, a deep study of the IoT system necessities is needed in order to conclude whether a high protection level is required or not. For instance, a factory where dangerous machines are being monitored for anomalous behaviour, and no data can be lost, high network resilience is mandatory, and, thus, high protection level. On the contrary, an IoT system with more relaxed necessities might prioritise energy consumption and, thus, deploy a WSN with low or even no protection level.

4.4 Conclusions

In this chapter, we have proposed a MIP formulation for the optimal placement of the WSN elements needed for sensing and acquiring the necessary information for building automation. In particular, the model guarantees optimal and minimal placement of sensors and gateways. Moreover, our model enhances the typical one presented in the literature by adding clustering constraints. Clusters are mandatory in future IoT systems due to the necessity to combine heterogeneous sensors inside the same network in order to sense and acquire all the required data. The definition of several cluster sizes allows for the different coverage needed by each type of sensor.

The model also fulfills connectivity, resource and protection constraints. Protection is another important aspect of WSNs, since network disruptions must not affect, in an extent, to the overall system. Protection has been further analysed with the help of a protection level constraint that defines the number of gateways a sensor must be able to reach. With this, the model ensures that, in the event of a partial network failure, sensors can redirect their transmission paths to the closest available gateway and guarantee that no data is lost. The study shows the energy impact of increasing such parameter to ensure high availability.

Obtained results show the capability of our proposal to reduce the amount of sensors and gateways needed for the deployment when compared with a complete scenario in which all the types of sensors are placed, if physically possible, in each room.

For instance, we need about 36% less sensors, and 64% less gateways than with respect to the case where all sensors and gateways are placed, with an overall energy savings of 38%. If protection is enforced, the number of sensors and gateways increases as well as the energy consumption. Nonetheless, with a protection level as high as 6, energy utilisation is still 18% better.

Chapter 5

Sensor Criticality Detection

Preface

This chapter presents and defines the sensor criticality study for homogeneous multi-hop WSNs. Homogeneous multi-hop networks are characterised by the utilisation of equal sensors across all the deployment, which have the ability to both sense and retransmit data generated by other sensors. Because of this, the utilisation of actual gateways is not mandatory. The equality these networks present obfuscates the most essential sensors of the network due to the inability to identify them. For this, we present a GRASP meta-heuristic for pinpointing the most important nodes inside the network. This identification allows for their enhancement with backup capabilities to increase, as much as possible, network availability, resilience and reliability.

5.1 Homogeneous Criticality Problem

As previously explained in Chapter 4, homogeneous networks still play an important role in future IoT systems. In order to extent current studies regarding this type of networks, we further analyse the problem of identifying the most essential sensors or *critical nodes* in a given network. It is important to state that the definition of node criticality varies from the common conception in the literature. A critical node is commonly defined as a cut vertex node, whose removal separates the network into two distinct parts unable to communicate.

In our solution, however, a critical node is defined as a node whose removal disrupts the network the most. In particular, such disruption is measured by two metrics, namely latency and lifetime, further explained in Section 5.1.1.

By identifying such sensors, it is possible to grant them with backup capabilities to avoid network disrupts by enhancing network resilience and availability.

5.1.1 Problem Statement

Given a WSN of N Mica2 nodes [85], represented as $V = \{1, 2, \ldots N\}$, a subset of critical node candidates $N_c \in V$, the number of critical nodes to remove C and a table T shown in Figure 4.1 of the possible transmission levels and the energy needed to transmit at each of these levels, a solution is found when each node i, apart from the Base Station and the critical nodes under testing, transmits to a node j at a transmission level k. Note that the chosen transmission level k is the least possible level that permits the interconnection between sensor i and j range-wise, in order to reduce energy consumption levels at the minimum possible ones.

Therefore, a solution is represented by two vectors, namely *send* and *key*, both of size N. The former stores, for each position i, the node j to which data is sent. Likewise, the latter stores the transmission level k needed for that communication. A solution is considered valid if $1 <= send_i <= N$ and $0 <= key_i <= |T| \; \forall i \in V \setminus N_c$.

Two different objective functions are considered, covering two of the major and most important requirements a WSN should have, latency and lifetime. The former makes reference to the required time to transmit data from one sensor to the sink node, whilst the latter refers to the amount of time the network can be maintained alive without needing to charge or change node batteries. We consider network lifetime as the summation of the hops needed from each sensor to reach the base station, as shown in Equation 5.1. The *calculatePathCost* returns such value for a given node and a given network state.

$$\sum_{n \in V} calculatePathCost(n, G_{current}) \tag{5.1}$$

Additionally, network lifetime is defined as the amount of packets that can be generated in each node until the critical one runs out of energy. It is worth noticing that this value is equal for all the nodes, and can be seen as the number of time slots t that the network is alive if, at each time slot, one packet is generated and transmitted in each of the network nodes apart from the base station.

$$Lifetime = \frac{battery}{E_{rx} * input + E_{tx} * output} \tag{5.2}$$

Equation 5.2 shows a general definition for lifetime, where *battery* is the amount of energy each node initially has, E_{rx} and E_{tx} are the values needed for receiving and transmitting a single packet respectively, and *input* and *output* are the number of connections each node sends and transmits, respectively.

In order to find the most influential subset of critical nodes of size C, it is needed to calculate the impact of each possible subset formed by the candidates. That is, given N_c, we compute all the possible combinations without repetitions $Perms$. Each of these combinations is then analysed to see its impact in terms of the current objective function under testing. Once all the solutions are found, nodes pertaining to the combination $Perms_i$ with the worst solution are considered the critical nodes.

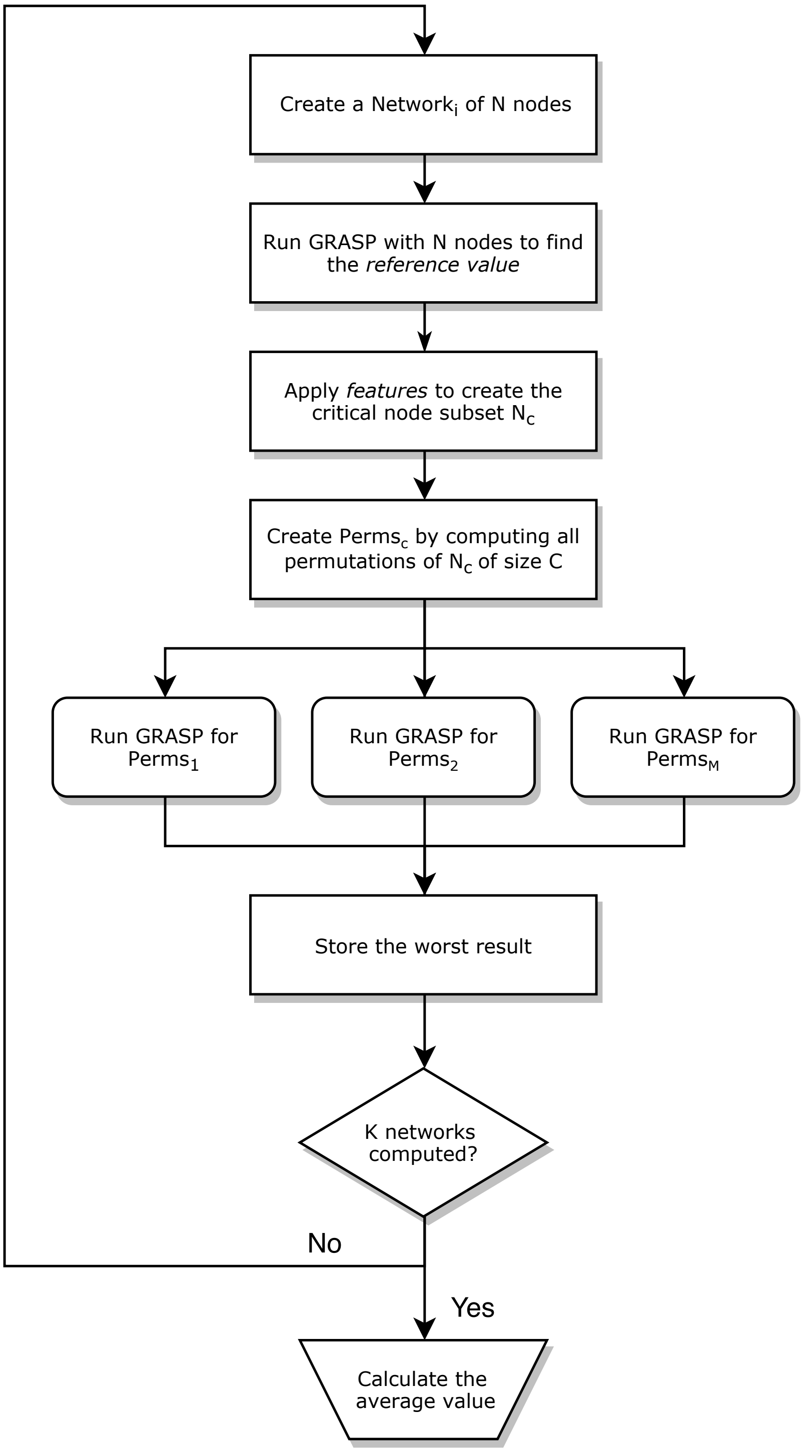

Create a Network$_i$ of N nodes
Run GRASP with N nodes to find the *reference value*
Apply *features* to create the critical node subset N$_c$
Create Perms$_c$ by computing all permutations of N$_c$ of size C
Run GRASP for Perms$_1$
Run GRASP for Perms$_2$
Run GRASP for Perms$_M$
Store the worst result
K networks computed?
No
Yes
Calculate the average value

Figure 5.1: Framework used for the identification of critical nodes

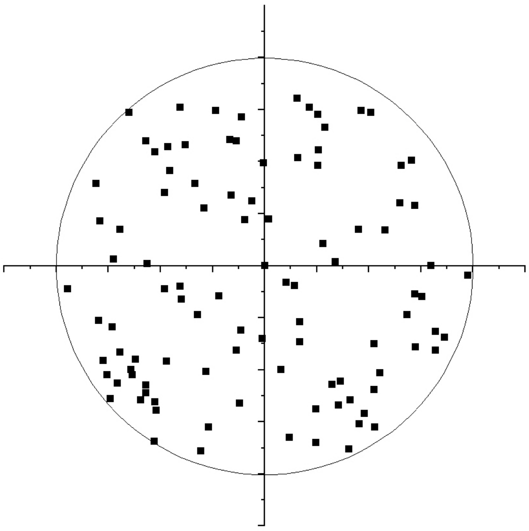


Figure 5.2: Example of the sensor distribution for $N = 100$ and $200m$ radius area.

Figure 5.1 shows the framework used for such analysis. Firstly, a new network of N nodes is created. This network is randomly generated by selecting N node coordinates within the radius range under study. Figure 5.2 shows an example of a network of $N = 100$ sensors distributed across a disk shaped area of $200m$ radius. After that, we initially run the GRASP without removing any node to store the reference value. In the event of not finding a valid solution due to the randomness of the network generation tool, the network is re-generated. This process is done iteratively until a valid network is found. Once it has finished, using the solution we perform a node selection by means of features that are later explained. From the result set of nodes, all the combinations are calculated. Since our purpose is to analyse the impact of critical nodes when they are simultaneously removed, it is worth mentioning that the features are important to reduce the number of $M = \binom{|N_c|}{C}$ combinations $Perms_c$ and thus, the complexity of the problem. For each of these combinations a GRASP is run and, as it can be seen in the figure, this is done in parallel. Once all the GRASPs have finished, the worst result is stored and a new network is analysed. When finishing the K valid networks, the average and maximum values are stored as results.

5.1.2 GRASP

GRASP (Greedy Randomised Adaptive Search Procedure) is a meta-heuristic proposed by Feo and Resende for solving hard computational problems [86]. The procedure is divided into two different phases, namely construction phase and search phase. In the former, a solution is greedily and iteratively constructed by adding one element to it until the size of the problem is reached. The latter phase is responsible for applying local search to the solution to try to improve it.

GRASP has been chosen due to its simplicity and high quality solution delivery in acceptable time. Such characteristic is key for our framework due to its necessity to run multiple instances of GRASP in order to check every permutation of the critical node subset, as shown in Figure 5.1. Additionally, the randomness presented in GRASP avoids local optimum solutions.

Latency GRASP

Algorithm 1 shows the generic code for the slightly tuned GRASP meta-heuristic used for the latency model. Regarding its input parameters, $Perms_i$ contains the identification of the current nodes under criticality test condition. T makes reference to the table of relationships between each transmission range and the energy needed to transmit at that level, as shown in Figure 4.1. $MaxIter$ defines the number of iterations to perform before finishing. α is a mandatory GRASP parameter that controls the level of randomness to be applied when creating a solution. It has been finally decided to use an $\alpha = 0.5$ for all the tests, since this value introduces a fair amount of randomness. Lower values might unable the GRASP to explore different neighbourhoods due to starting most of the times from the same initial solution, and higher values might create bad initial solutions that the local search is then unable to properly improve.

Finally, G specifies a valid structure of a solution. As an output, the best solution found in terms of the current cost function being used is returned.

As for the actual code, line 2 contains the creation of an initial *random* solution to the problem. Then, this solution is improved by applying a local search algorithm as specified in line 3. Finally, if the cost of the new solution is better than the momentary best, the solution is updated.

Algorithm 1 GRASP($G, Perms_i, T, MaxIter, \alpha$)

```
Require:
  Perms_i: subset of critical nodes to remove
  T: transmission levels
  α: randomness level specific for GRASP
  G: node graph
Ensure:
  G_best: new node graph
1: while (MaxIter > 0) do
2:    $G_{current} \leftarrow$ greedyRandomisedSolution($\alpha, Perms_i$)
3:    $G_{current} \leftarrow$ localSearch($G_{current}, maxNoImprovIter$)
4:    if (cost($G_{current}$) < cost($G_{best}$)) then
5:       $G_{best} \leftarrow G_{current}$
6:    end if
7:    MaxIter = MaxIter - 1
8: end while
9: return $G_{best}$
```

As it can be seen, in order to define the GRASP meta-heuristic, the greedy solution construction and the local search methods need to be defined. Following sections show both of these methods in more detail.

Construction Phase

During the first step of the GRASP, namely construction phase, a valid solution must be built. Algorithm 2 shows the code of the method responsible for such matter. Starting from an empty set as shown in line 1, we define a loop in lines 2-16 in which each iteration is responsible for adding a new node to the solution until the size of the problem is reached. To decide which node to include in the current step, firstly, the cost of adding each node outside both the partial solution and the critical nodes subset $Perms_i$ is computed, as seen in lines 4-6. When all the costs are calculated, a Restricted Candidate List (RCL) is built by adding only the nodes with a cost inside the range [C_{min}, α * (C_{max} - C_{min})] (lines 10-14). These thresholds represent the minimum and the maximum *adding* cost for the nodes yet outside the solution. To finalize, a random node is extracted from the RCL and added to the solution.

The objective function of the problem is defined inside the function responsible for calculating the cost of adding an element to the current solution, as seen in line 5.

Algorithm 2 greedyRandomisedSolution($\alpha, Perms_i$)

```
Require:
   α: randomness level specific for GRASP
Ensure:
   G: node graph
 1: $G \leftarrow \emptyset$
 2: while (|G| < problemSize) do
 3:    $Node_{costs} \leftarrow \emptyset$
 4:    for $Node_i \notin G$ and $Node_i \notin Perms_i$ do
 5:       $Costs_i \leftarrow$ costOfAdding($Node_i$, $G$)
 6:    end for
 7:    $RCL \leftarrow \emptyset$
 8:    $C_{min} \leftarrow$ minCost($Costs$)
 9:    $C_{max} \leftarrow$ maxCost($Costs$)
10:    for $Cost_i \in Costs$ do
11:       if $(Cost_i \leq C_{min} + \alpha * (C_{max} - C_{min}))$ then
12:          $RCL \leftarrow Node_i$
13:       end if
14:    end for
15:    $G \leftarrow$ selectRandomNode($RCL$)
16: end while
17: return G
```

In this construction phase, there is no order in which nodes are added to the solution, neither a global condition to test if the insertion is the optimal one. That is why the method is considered greedy, because in each step, the best local cost is chosen as a candidate.

Search Phase

The second phase, namely search phase, of the GRASP algorithm is responsible for improving, if possible, the solution created by the construction phase. In order to do so, the search explores the solution neighbourhood to check if there are any with better cost in terms of the objective function under use.

Algorithm 3 shows the code for the local search. Regarding its parameters, $G_{current}$ contains the solution to be improved and $maxNoImprovIter$ specifies the value of the stopping criteria. It limits the number of consecutive iterations to perform without solution improvement. Once this threshold is reached, we consider that $G_{improved}$ contains the local best.

For improving the solution, the algorithm firstly gets a *random* node n from the solution. Then, it computes the current cost of sending data through its path until reaching the base station. Since this path has been greedily selected by the constructive phase, we check whether there are alternative paths through any of the neighbours of n ending at the base station that has better cost. If so, the path is updated and the local search continues. Also, the consecutive iterations without improvement are reset.

Algorithm 3 localSearch($G_{current}, maxNoImprovIter$)

```
Require:
  $G_{current}$: current solution
  $maxNoImprovIter$: maximum number of consecutive iterations to perform without improvement
Ensure:
  $G_{improved}$: improved solution
stop = 0
while $stop < maxNoImprovIter$ do
    $n$ = getRandomNode($G_{current}$)
    $best$ = calculatePathCost($n, G_{current}$)
    for $m \in$ Neighbours($n$) do
        $cur$ = cost($n, m$) + calculatePathCost($m, G_{current}$)
        update($G_{current}$)
        if $cur < best$ then
            $stop = 0$
            $G_{improved} \leftarrow G_{current}$
        end if
    end for
    $stop = stop + 1$
end while
return $G_{improved}$
```

Lifetime GRASP

The lifetime GRASP code is very similar to the latency one presented in Section 5.1.2 thanks to the inner characteristics of the meta-heuristics, which deliver a structure that needs very small tuning for solving different problems.

The outer GRASP code presented in Algorithm 1 is valid for the lifetime model since it also needs the critical node subset in each execution to check the one with more lifetime impact. Moreover, the greedy randomised method previously presented can also be re-utilised. In the case of the local search, major changes have been performed in order to correctly explore the neighbourhood and improve the solution as much as possible.

Following sections explain more in detail each of the methods and the new decisions taken.

Construction Phase

As it has been said, the construction phase for the lifetime model follows the exact same approach as the latency model. Algorithm 2 is, therefore, also valid for this model. The reason behind such decision relies in the fact that the more hops a solution contain, the more energy is consumed and, thus, lifetime is not as high as it could have possibly been. For calculating the overall network lifetime, lets consider a node i, with maximum battery capacity of $3J$. Let f_{ij} be the amount of connections that node i redirects to node j. Since every node only generates its own connection, it can be rapidly seen that $f_{ij} - 1$ is the amount of connections that node i receives. The following formula calculates the lifetime, or amount of time steps t that node n can hold up without rendering out of energy:

$$Lifetime = \frac{3 * 10^9}{922 * (f_{ij} - 1) + T_{ij} * f_{ij}} \tag{5.3}$$

Where $922\,nJ$ stands for the amount of energy needed for receiving one packet, and T_{ij} makes reference to the amount of energy needed to transmit from node i to node j according to their distance, as specified in Table 4.1.

The formula is divided into two parts: energy needed to receive and energy needed to retransmit. Since the energy needed to receive can be seen as an overhead, we concluded that a network with minimum hop count holds the lowest possible retransmissions and, thus, the additional overhead paid for having to receive packets is minimal. In some cases, this statement could be wrong: according to the energy and range values seen in Table 4.1, in very specific scenarios, it can be calculated that the amount of energy for doing two hops plus the energy needed to receive allows the network to reach the same distance with lower energy consumption that one single hop. However, this constructive algorithm provides a very good starting point and the local search is responsible for improving it by correcting such scenarios.

The overall network lifetime is the minimum among all the node lifetimes since this value represents the amount of time steps until the first node exhausts its battery.

Search Phase

Search phase for the lifetime model follows the same pattern as the latency one, but with completely different swapping criteria. Algorithm 4 shows the method responsible for improving the input solution. As it can be seen, the parameters are exactly the same as before, being $G_{current}$ a valid constructed solution and $maxNoImprovIter$ the maximum number of iterations without improvement until the search finishes.

Due to the heterogeneity on the energy needed to transmit depending on the range and the different relay that each network node holds, it is safe to say that there will be a node or a group of nodes restricting the lifetime. Then, the local search is responsible for finding the congested nodes and balancing their load in order to increase the overall network lifetime. To do so, we firstly identify the more congested node c. In order to reduce its load, we randomly select one node n sending data to such congested node c. Then, we try to redirect the connectivity of n to another of its neighbours different from c so that the network load is more distributed and the lifetime can be increased. To avoid making changes with which the lifetime is not increased, we only mark the current change as possible if the lifetime l can be increased at least to $l + 1$. If the conditions are met, the network is updated for the following iteration.

Algorithm 4 localSearch($G_{current}, maxNoImprovIter$)

```
Require:
  $G_{current}$: current solution
  $maxNoImprovIter$: maximum number of consecutive iterations to perform without improvement
Ensure:
  $G_{improved}$: improved solution
$stop = 0$
while $stop < maxNoImprovIter$ do
    $c$ = getCongestedNode($G_{current}$)
    $n$ = getRandomLeaf($G_{current}, c$)
    $lifetime$ = calculateMaxLifetime($G_{current}$)
    for $m \in$ Neighbours($n$) do
        if changeIsPossible($G_{current}, n, m, lifetime$) then
            $G_{improved}$ = makeChange($G_{current}, n, m$)
            $newLifetime$ = calculateMaxLifetime($G_{improved}$)
            if $newLifetime > lifetime$ then
                $G_{current} = G_{improved}$
            end if
        end if
    end for
    $stop = stop + 1$
end while
return $G_{current}$
```

Feature Selection

Given a WSN with N nodes, it is safe to say that node importance inside such network significantly varies from one to another. Since all the nodes must finally transmit their data to the sink node (i.e. base station), it can be seen that the furthest nodes will act as *leaves* by only sending data, without needing to retransmit another node's packets. However, nodes very close to the base station will most likely act as bridge nodes, retransmitting the data from the rest of the nodes that are unable to reach the base station by themselves. Having this in mind, we have defined 3 features that allow us to rank the importance of the nodes in order to reduce the number of criticality tests. These features are the following:

Connectivity Given a node $i \in N$, node connectivity is defined as $C = |\{\, j \,|\, j \neq i, distance(i,j) \leq T_{range}^{max} \,\}|$ where T_{range}^{max} refers to the maximum possible range that a node can transmit using the most costly transmission level shown in Table 4.1.

Hops to base station For a node i, this feature is defined, as its name indicates, as $H = \frac{distance(i,\, bs)}{T_{range}^{max}}$. Where bs stands for base station and T_{range}^{max} refers to the maximum possible range that a node can transmit using the most costly transmission level shown in Table 4.1.

These 2 initial features can be calculated prior to obtaining any solution for a given network, since the only requirement for calculating them is the distance between nodes, which can be extracted from the node coordinates. However, we also define another feature directly related to the solution of the base case scenario (i.e. when no critical nodes are removed from the network):

Relay A node i defines its relay as the number of outgoing connections. This number is equal to the amount of ingoing connections plus its own generated ones. Given the solution matrix f and the value f_{ij} which indicates how much data node i sends to node j, the relay for node i is defined as $R = \sum_{j \in N} f_{ij}$.

Once all the features are defined, it is needed to see whether their use actually allow us to reduce the number of nodes to test for criticality without losing precision in the solution. In order to do so, 10 network topologies of $N = 100$ nodes are created by using a random node distribution in a $200m$ radius area. These topologies are then analysed using the GRASP meta-heuristic presented in Section 5.1.2 for obtaining the identification of the critical nodes by testing all the possible nodes and subset of nodes for both the latency and the lifetime models. Once the critical nodes are identified, we search their ranking position in the three features. It is worth mentioning that the nodes pinpointed as critical may actually not be the correct ones obtained by the analogous MIP model. However, we firstly analyse the features by only checking the GRASP, and, in Section 5.2, we analyse the percentage of identification success.

#Crit Nodes	H-1	Relay-3	Relay-5	Relay-10	Conn-3	Conn-5	Conn-10
1	100%	70%	70%	90%	0%	20%	30%
2	100%	45%	55%	80%	5%	25%	25%

Table 5.1: Independent feature success rate for top 3, 5 and 10 thresholds for the latency model. $H = 1$ is enough for 100% success rate.

Tables 5.1 and 5.2 show the overall percentage of success for each critical node subset tested and different individual feature thresholds for the latency and the lifetime models respectively. Specifically, top 3, 5 and 10 are used for both connectivity and relay. The hops

#Crit Nodes	H-1	H-2	Relay-3	Relay-5	Relay-10	Conn-3	Conn-5	Conn-10
1	90%	100%	20%	40%	100%	40%	60%	70%
2	90%	100%	15%	35%	100%	25%	45%	70%

Table 5.2: Independent feature success rate for top 3, 5 and 10 thresholds for the lifetime model.

to base station, however, differ from the two tests: in the case of the latency, it is sufficient to have $H = 1$ in order to achieve a 100% success. The lifetime model needs to extend this range up to $H = 2$ to achieve the same result.

As can be seen, the hops feature is a really good critical node predictor, since all the critical nodes are always very close to the base station. Relay is also a good predictor, since the values obtained for the *top 10* are very high in both scenarios. However, connectivity does not seem to offer good reliability to identify critical nodes.

With these results, it is clearly seen that enforcing high rank in all the features to a node in order to include it in the critical node subset is not reliable and the percentage of success will be very low. Similarly, one may think that testing all the nodes with $H = 1$ for the latency or $H = 2$ for the lifetime will be the most reliable scenarios. Nevertheless, by only restricting such feature, the number of nodes for testing is, on average, still very high to obtain plausible times when running GRASP for larger networks and larger subsets of critical nodes.

For all of this, it has been decided to enforce a membership of two out of the three features, independently of the combination, to consider a node plausible for criticality testing. Tables 5.3 and 5.4 show the results of such scenarios. As it can be seen, the identification of the most critical node increases when increasing the top threshold, and it arrives to 100% when the *top 10* is used. In the case of 2 critical nodes, the percentages generally decrease. However, using the same *top 10* threshold, the success rate is considerably high (80%) for the latency, and achieves a 100% success rate in the case of the lifetime.

#Crit Nodes	Top 3	Top 5	Top 10
1	70%	90%	100%
2	50%	65%	80%

Table 5.3: Critical node identification success percentage when utilising different top rank thresholds for relay and connectivity for the latency model. $H = 1$ is fixed.

#Crit Nodes	Top 3	Top 5	Top 10
1	60%	70%	100%
2	55%	75%	100%

Table 5.4: Critical node identification success percentage when utilising different top rank thresholds for relay and connectivity for the lifetime model.

In conclusion, features are a good tool for reducing the complexity of the problem by selecting only candidates with high possibility of being critical instead of testing every single network node. Results presented in Section 5.2 are extracted by utilising such features. If nothing is indicated, results are extracted using *top 10* for both connectivity and relay, whilst $H = 1$ is used for hops to base station in both models, in order to maintain consistency.

5.2 Results

This section presents the node criticality study results for WSNs. As it has been previously explained, the definition of node criticality varies from the common conception in the literature. A critical node is commonly defined as a cut vertex node, whose removal from the network separates it in two parts unable to communicate.

In our solution, however, a critical node is defined as a node whose removal disrupts the network the most. In particular, two main WSN metrics are taken into account for discovering critical nodes, namely latency and lifetime. Remind that our solution considers latency as the overall number of hops needed to transmit data from each sensor up to the sink node, and lifetime is calculated as the number of time intervals or slots until the first node exhausts its battery.

Since the presented solution comprises the usage of a meta-heuristic which might not give the optimal solution, it is needed to firstly validate the model to see whether it actually delivers close to optimal solutions. To do so, a comparison between the MIP results and our results is presented for small networks. Results are obtained by studying 10 networks of $N = 100$ nodes evenly distributed in a $200m$ radius area. For each of these networks, the latency and lifetime values for both algorithms as well as the nodes pinpointed as critical are shown. Since the MIP model offers optimal results, the comparison allows us to evaluate the effectiveness of the GRASP model.

Then, the study of larger networks is initially performed by distributing up to $N = 500$ nodes in the same $200m$ radius area. Due to the limitation in the area under study, it can be noted that, as the number of nodes increases, so it does the node density. Thus, the variation in the metric under study should decrease with the node increment.

For $N = \{100, 200\}$ network nodes, results are shown for up to 5 simultaneous critical node removal. In the case of $N = 300$, results are reduced to 4 critical nodes. Finally, for $N = 500$, the 3 most influential nodes are calculated.

Networks with more node density have not been studied because of the already small values obtained during the aforementioned tests, and, since GRASP is not an exact tool, it is difficult to extract correct conclusions with such little margin error. Moreover, the limitation in the area to deploy the sensors highly increases the complexity of the problem due to the amount of neighbours each node can send data to, increasing the explorable neighbourhood during construction and search phases.

Number of Nodes	Radius	Hops to base station
100	200	1
300	350	2
500	450	2.5
700	530	3
1000	630	3.5

Table 5.5: Larger network parameters scaled from base $N = 100$ node network.

For all that, it has been decided to scale the initial $N = 100$ and $200m$ radius network in order to study larger ones without varying the initial node density. Moreover, the *hops to base station* threshold feature is tuned to also consider this scaling. Table 5.5 shows the radius and hops to base station thresholds used for networks with $N = \{100, 300, 500, 700, 1000\}$ nodes. The rest of the feature thresholds remain the same. Since these new networks share the same characteristics in terms of node density and feature thresholds, but scaled up, we want to see whether the impact of critical nodes also remains constant through all the cases. The decision for not studying the impact of more critical nodes relies in the trade-off between the time needed for finding the solution and the network degradation variation.

	MIP	GRASP
100 Nodes	2 min	5.4 seconds

Table 5.6: Comparison of the execution times for the MIP model and the GRASP meta-heuristic for $N = 100$ nodes and 1 critical node removal.

Table 5.6 shows a comparison between the MIP and GRASP execution times needed for selecting the most critical node in a network of $N = 100$ nodes. As it can be seen, the difference in time is substantial, which is the main reason to create the meta-heuristic for obtaining results for bigger networks and wider set of critical nodes.

5.2.1 Latency Results

This section presents the results for the GRASP meta-heuristic with latency as the metric under evaluation. Remind that latency is calculated as the overall number of hops needed for each node th reach the network sink.

Table 5.7 summarises the results obtained for 10 different random network configurations. Particularly, it compares the latency value obtained for both the MIP and the GRASP, as well as the identifier of the nodes pinpointed as the most critical ones. Each network case is summarised into two consecutive rows. The first row makes reference to the results of the most critical node removal, whilst the second one represents the removal of the two most critical nodes. As previously stated, the critical nodes in the MIP model are iteratively extracted. For this reason, each row contains, respectively, the first and second most critical node. In the case of the GRASP, however, nodes are removed concurrently. Due to this, each row represents an independent result and complete results are shown for the most and the combination of the two most critical nodes.

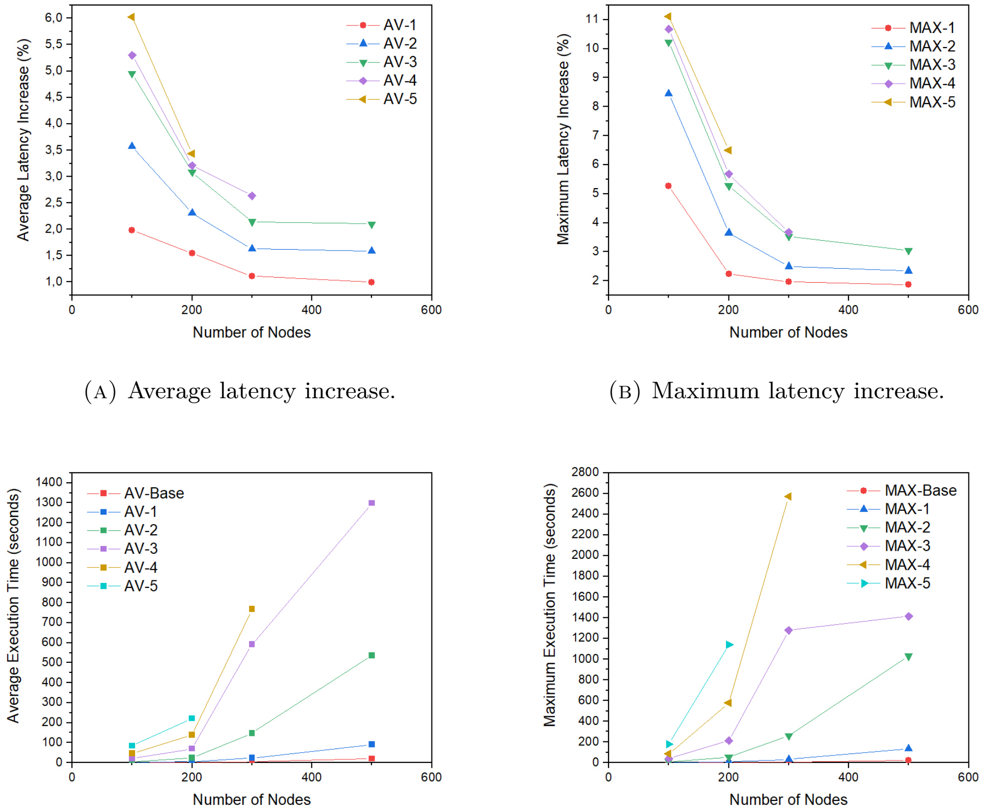
	Latency Values			**Critical Node Ids**		
	MIP	GRASP	Gap	MIP	GRASP	Success Rate
Network #1	235	235	0%	26	26	100%
	236	236	0%	59	26, 59	100%
Network #2	240	240	0%	31	31	100%
	242	243	0.41%	(60, 84)	31, 60	100%
Network #3	231	233	0.86%	70	70	100%
	234	236	0.85%	85	16, 70	50%
Network #4	211	211	0%	(20, 21)	20	100%
	211	212	0.47%	(32, 77, 86)	20, (32, 77)	100%
Network #5	214	216	0.93%	23	23	100%
	217	219	0.92%	(43, 92)	23, 55	50%
Network #6	241	242	0.41%	(12, 81)	81	100%
	244	245	0.4%	(12, 25, 81)	25, 39	50%
Network #7	231	232	0.43%	2	2	100%
	234	235	0.42%	(53, 91)	19, 27	0%
Network #8	248	248	0%	30	98	0%
	249	250	0.4%	(24, 98)	43, 98	50%
Network #9	228	229	0.44%	96	96	100%
	234	235	0.42%	77	64, 96	50%
Network #10	233	233	0%	17	17	100%
	242	244	0.82%	96	17, 96	100%
Average			0.307%			90%
			0.511%			65%

Table 5.7: Comparison between MIP and GRASP latency models. Nodes between parenthesis are interchangeable since they offer the same result when being removed.

Regarding the identification of the critical nodes, GRASP offers really good results when extracting the most influential node, delivering a 90% success rate. However, when testing the network for the two most critical nodes, this rate is reduced to around 65% due to the GRASP failing to identify one or both of the critical nodes. The justification of such low value lies in the fact that MIP and GRASP tests for this scenario slightly differ. Since GRASP nodes are being removed simultaneously instead of iteratively, there might be a combination of two nodes that deteriorates the network more than the iterative removal of the two most influential ones.

Network #8 is further analysed in order to understand why the first critical node is missed. The first important aspect to notice is that the missed and the correct critical nodes are inside the critical node test set selected by the features. Specifically, both are in range of the base station and their neighbourhood is inside the top 10. Regarding the solution obtained by GRASP, the relay of both nodes slightly differ. However, it can be seen that both send data directly to the base station. The MIP results deliver the same scenario, but in this case the relay is very similar between them.

After seeing that the characteristics of the missed and the critical nodes are almost identical, a conclusion can be extracted by looking at the latency results. The difference of the MIP latency values obtained for the two nodes is minimal, making it difficult for the GRASP to actually find the optimal value, and instead, obtaining a slightly worse one with a different node pinpointed as critical.



(A) Average latency increase.

(B) Maximum latency increase.

(C) Average latency execution times in seconds.

(D) Maximum latency execution times in seconds.

FIGURE 5.3: Average and maximum latency increase values and execution times for networks with nodes distributed in a 200m radius area.

Figure 5.3a depicts the results of the previously explained tests in which node density is increased by adding more nodes to the same network deployment area of 200m. Each of the lines drawn A-C shows the average impact when removing the most C critical nodes from the network.

There are two main conclusions that can be extracted. Firstly, it can be clearly seen that as the number of nodes inside the network increases, its deterioration when removing a fixed quantity of critical nodes decreases. As there are more nodes inside the network, the paths to which a node can transmit increase, the connections between the nodes are more distributed and, thus, network flexibility increases, reducing the impact of critical node removal. Specifically, it can be seen that latency deterioration in 500 node networks is half that of 100 node networks.

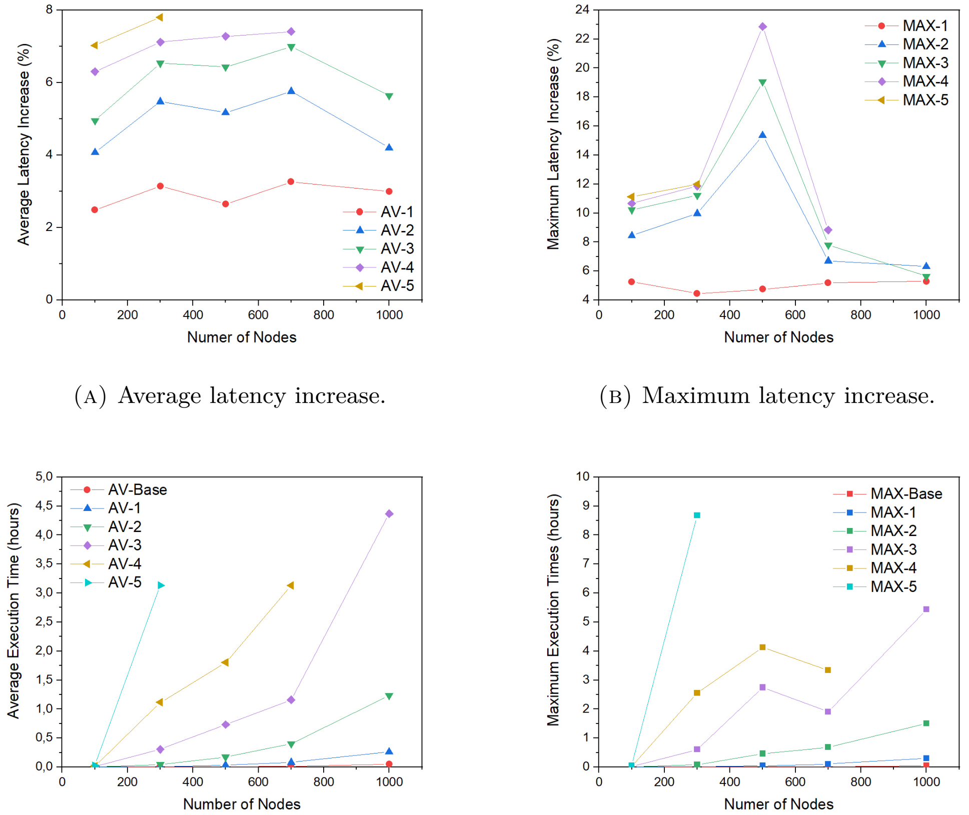
Secondly, when looking at the values of a single network size, it can be seen that the removal of more critical nodes has higher impact on the latency increment. The theory behind such result is the following: let c be the critical node removed and $Parents_c$ the nodes that send data to it. When c is removed, latency is reduced by the amount of hops c needs to send data to base station but, at the same time, it increases by at least $|Parents_c|$ since the new path for such nodes will be at minimum 1 hop larger. If happens to be a faster path, it would have been selected either during construction or search phase. This explanation is not true when c shares two paths with exact same length or the network does not have inner nodes. However, such scenarios are very specific and only possible in networks with very few nodes, which is not the case. The explanation is also extensible when more than 1 critical nodes are removed. It is worth mentioning that these are theoretical calculations and they might slightly differ from the obtained results in some cases due to the inexactitudes of the GRASP meta-heuristic.

Additionally, in order to clearly see how much impact the removal of C critical nodes can have in a network, the maximum values are shown in Figure 5.3b. Even though the tendency of the curves remains almost the same as before, the latency increment is much higher, arriving to double the average in some cases.

Figures 5.3c and 5.3d show, respectively, the average and maximum execution time needed for calculating the previous values. As it can be seen, the time needed to remove a fixed amount of critical nodes increases exponentially with the number of nodes, mainly due to the amount of neighbourhood checks to perform. Moreover, when looking at a fixed network size, it can also be seen that the tendency when removing a higher amount of critical nodes is exponential because of the combinatorial factor introduced for calculating all possible combinations of critical nodes subsets.

Figure 5.4a shows the average results after scaling the $N = 100$ node network deployed within a $200m$ radius area. Performing a scaling to keep the same characteristics as the $N = 100$ and $200m$ radius network allows us to conclude that the latency deterioration remains the same, with very small variation, when removing the same amount of critical nodes.

However, when looking at the maximum results in Figure 5.4b, the horizontal tendency disappears. Even though the average is maintained, it can be seen that the worst scenarios can have huge deterioration variations. For instance, the worst network of $N = 500$ nodes increases its latency by 25% when removing more than 4 critical nodes, and this percentage is drastically reduced to around 10% for the $N = 700$ network. This results are representative of the network space studied, and since these values are worst cases, the study of more networks could possibly change them. However, due to the number of nodes and distribution area, it is infeasible to cover all possible configurations.



(A) Average latency increase.

(B) Maximum latency increase.

(C) Average latency execution times in hours.

(D) Maximum latency execution times in hours.

FIGURE 5.4: Average and maximum latency increase values and execution times for networks with nodes distributed according to area specified in Table 5.5.

Even though scaled networks perform differently in terms of latency deterioration, the exponential trend in the execution time remains the same as shown in Figures 5.4c and 5.4d.

5.2.2 Lifetime Results

Equally to the latency model, the comparison between the MIP model and the GRASP meta-heuristic for small networks of $N = 100$ nodes distributed in a $200m$ radius area is firstly presented. These results allow us to appreciate the margin error that the GRASP offers. Moreover, it can be checked whether GRASP ensures a correct critical node identification. Table 5.8 shows such results. As in the latency case, each network case is summarised into two consecutive rows. The first row makes reference to the results of the most critical node removal, whilst the second one represents the removal of the two most critical nodes. As previously stated, the critical nodes in the MIP model are iteratively extracted. For this reason, each row contains, respectively, the first and second most critical node. In the case of the GRASP, however, nodes are removed concurrently. Due to this, each row represents an independent result and complete results are shown for the most and the combination of the two most critical nodes.

	Latency Values			**Critical Node Ids**		
	MIP	GRASP	Gap	MIP	GRASP	Success Rate
Network #1	167,909	120,252	28.38%	53	53	100%
	145,844	99,595	31.71%	13	30, 53	50%
Network #2	85,027	75,414	11.3%	84	(3, 15, 84, 60)	100%
	85,027	75,414	11.3%	60	60, 84	100%
Network #3	88,353	75,414	14.64%	85	85	100%
	75,438	70,277	6.84%	25	55, 85	50%
Network #4	301,263	191,629	36.39%	57	57	100%
	272,910	154,411	43.42%	83	57, 87	50%
Network #5	348,441	283,650	18.59%	44	32	0%
	328,744	257,812	21.57%	87	32, 87	50%
Network #6	118,880	118,842	0.03%	32	32	100%
	88,353	74,129	16.1%	36	36, 89	50%
Network #7	220,426	196,592	10.81%	91	36	0%
	134,354	100,277	25.36%	36	36, 40	50%
Network #8	153,052	118,842	22.35%	16	16	100%
	121,982	88,053	27.81%	11	16, 98	50%
Network #9	220,427	154,411	29.94%	96	96	100%
	181,638	131,809	27.43%	97	38, 96	50%
Network #10	105,355	89,298	15.24%	100	100	100%
	97,318	89,298	8.24%	58	58, 100	100%
Average			18.76%			80%
			21.97%			60%

TABLE 5.8: Comparison between MIP and GRASP lifetime models. Nodes between parenthesis are interchangeable since they offer the same result when being removed.

Table 5.8 shows the comparison between the MIP model and the GRASP results, as well as the pinpointed critical nodes. The lifetime values shown represent, as previously explained, the *time slots* the network can be maintained fully operative if every node generates and sends one packet in each of these *time slots* until the first one runs out of battery.

In this case, the difference between the MIP and the GRASP substantially differ. Thanks to the exactness of the MIP, node transmissions are split between different nodes instead of sending all the data to a single one. However, due to the approximation behaviour of GRASP, the split of the transmissions cannot be exactly performed. As it can be seen, GRASP lifetime is 18% less for the 1 critical node removal case, and it increases to around 22% for the two most critical nodes tests. Looking at the networks independently, it can be seen that, in some cases, GRASP almost reaches the best value given by the MIP model.

Critical node identification also suffers a degradation compared to the latency model. When removing the most critical network node, success rate is lowered to 80%, mainly due to the efficient transmission split that MIP performs. In the case of the two most critical nodes removal, the difference in which nodes are removed from the network also lowers the success rate to 60%.

Networks #5 and #7 have been further analysed in order to understand why the first critical node is not correctly pinpointed. By looking at the feature rank and individual results that both the MIP and the GRASP deliver for the correct and the missed nodes, it can be clearly seen that such nodes share many similarities, being their constant appearance in the critical node test set the first and most important one.

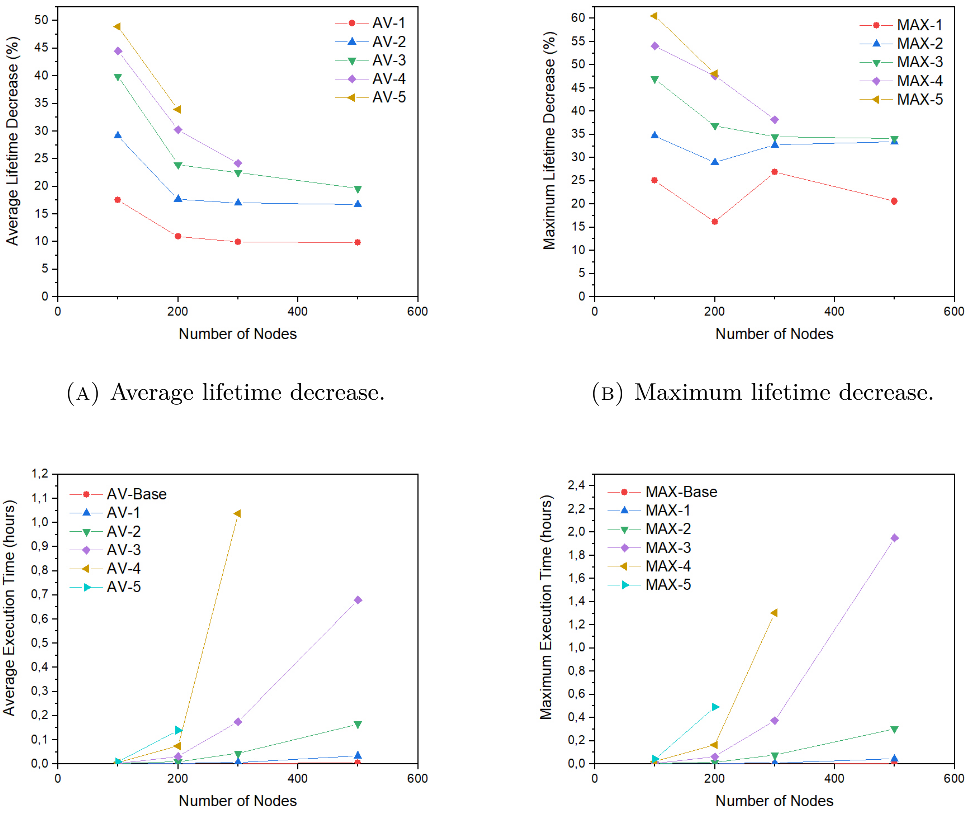
In the case of network #5, the missed and the correct critical nodes are at reach of the base station, with an extensive neighbourhood. Regarding the relay feature, the MIP grants both nodes a very similar value, among the top ones, and the GRASP assigns the same relay value. Equally to the latency case mentioned before, the similarity in the final MIP lifetime value makes it difficult for the GRASP to properly find the optimal value.

Network #7 share many similarities with the aforementioned network #5. Equally, the missed and the correct critical nodes are in range of the base station, and they both send data to it. Moreover, their neighbourhood is among the top 10. Moreover, the MIP relay values are almost identical between the two nodes, being identical in the case of the GRASP. Final lifetime values only differ by a single unit in the case of the MIP. For such reason, it is very difficult for the GRASP to reach the optimal scenario.

In conclusion, even though the success rate for the first critical node is not 100%, it can be seen that the missed cases can also be useful for understanding the characteristics of the network and the critical nodes.

Figure 5.5a shows the results for the tests in which node density is increased. Each of the lines A-C shows the average impact when removing the most C critical nodes from the network. As it can be seen, the behaviour of the previous average latency increase results and the current average lifetime decrease share many similarities. Firstly, as the number of nodes inside the network increases, the degradation in the lifetime lowers, mainly due to the flexibility that dense networks offer in terms of available paths for distributing the data delivery. Second and most importantly, it can be seen that if nodes inside the network share the same battery capacity, the lifetime decreases as the number of critical nodes removed increases. When a critical node n is removed, the network slightly loses flexibility, and paths crossing n need to be redirected to a more costly destination, reducing the lifetime of those nodes and thus, the overall network lifetime.

Figure 5.5b extends the previous results by showing, in this case, the maximum lifetime decrease values obtained. It is worth noticing that although the decreasing tendency is slightly maintained, the smoothness of the curve previously seen disappears. Again, these results are only representative of the network space tested, and the study of different network layouts may vary the maximum results. However, it is not possible to study every single network due to the amount of possibilities.



(A) Average lifetime decrease.

(B) Maximum lifetime decrease.

(C) Average lifetime execution times in hours.

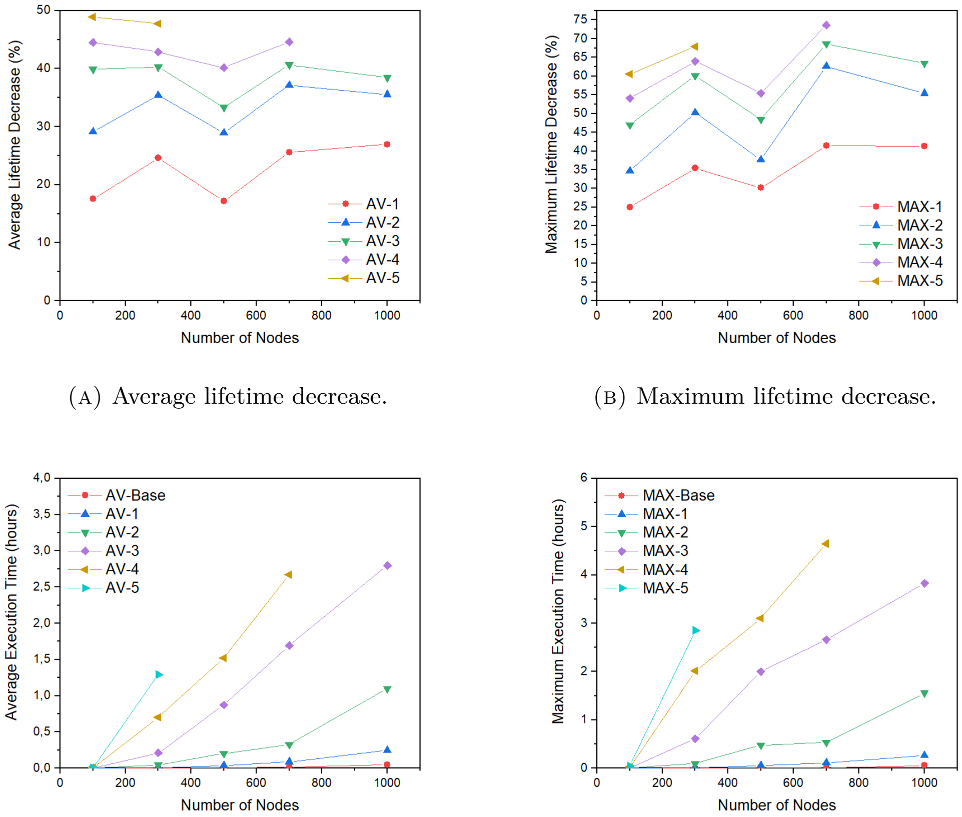
(D) Maximum lifetime execution times in hours.

FIGURE 5.5: Average and maximum lifetime decrease values and execution times for networks with nodes distributed in a 200m radius area.

Average and maximum execution times are respectively shown in Figures 5.5c and 5.5d. Both plots show the exact same exponential tendency. However, maximum values are much higher, arriving to double the average in some cases.

Results for lifetime in scaled networks are presented in Figure 5.6. Similar to the latency model, the horizontal tendency appears as shown in Figure 5.6a. However, in this case the variation between different network sizes are more considerable but it can be concluded that the lifetime decrease is maintained when the network is scaled up. Nonetheless, when looking at the maximum decrease values in Figure 5.6b, instead of a horizontal tendency, a slight increase is seen.

Figure 5.5d shows the average execution time for the lifetime decrease model. The exponential tendency is clearly seen for the AV-2, AV-3 and AV-4 cases. Equally, in the case of maximum execution times depicted in Figure 5.6d, this exponential tendency continues but with bigger time values, arriving to more than 2 hours in some cases.



(A) Average lifetime decrease.

(B) Maximum lifetime decrease.

(C) Average lifetime execution times in hours.

(D) Maximum lifetime execution times in hours.

FIGURE 5.6: Average and maximum lifetime decrease values and execution times for networks with nodes distributed according to area specified in Table 5.5.

5.3 Conclusions

In this work, we have extended the results obtained by the control MIP model presented in [44, 47] thanks to a GRASP meta-heuristic capable of handling larger networks in terms of nodes and also larger tests in terms of number of critical nodes. Moreover, the decision of simultaneously removing such critical nodes instead of doing it iteratively has driven us to the following conclusions:

Simultaneous removal The simultaneous removal allows for covering the worst possible scenario, in which part of the network unexpectedly shuts down and many critical nodes are disabled at the same time. An IoT system distributed across many buildings or zones can have part of the network disabled due to natural disasters, malicious attacks or power outages. On the contrary, individual sensor incapacity is less likely to happen due to the possibility to monitor and prevent unwanted battery levels or sensor behaviours prior to the actual failure.

Flexibility In many cases, the iterative and simultaneous cases end up identifying the same nodes as critical. This means that if, for any reason such as time limitation or computational power, it is not possible to cover the simultaneous case, by computing the iterative scenario, it is enough to identify the most disruptive nodes with sufficient certainty. However, this should not be the common proceeding.

Nonetheless, by looking at the results presented, we can appreciate that the identification of critical nodes is crucial to deliver good service even when the network is being disrupted. Although their impact on fixed deployment areas decreases as the number of nodes increases, their failure is still relevant enough to consider securing their integrity. Moreover, as deployment area scales, the impact of the critical nodes remains almost equal, contradicting the idea that, since there are more nodes, their importance should be lower as the number of nodes increases. Because of that, it is clear that the security and reliability of the most influential nodes is very important to avoid the shut-down of a whole system network, which can induce large economic losses for a company that deeply relies on gathered data to acquire benefits or perform its duty correctly.

Additionally, even though meta-heuristics do not deliver exact and correct results all the times, the success rate obtained offers a good opportunity for large scale scenarios in which exact optimisation tools are unable to handle such amount of data.

Chapter 6

Building Energy Management System

Preface

This chapter presents the definition and parametrisation of a smart building automation use case. The architecture presented in Section 3.2 is used as the basis to define and construct all the required elements to undertake the solving of the use case. Due to the elevated cost of a real scenario, building behaviour and element status are simulated, as well as people's movements and interactions. The aim of the simulator is to detect whether it is possible to optimise energy consumption whilst maintaining acceptable levels of occupant comfort. For this, energy consumption and comfort is calculated throughout the simulation. Moreover, such trade-off is studied by means of three distinct scenarios, in which building layout and number of occupants varies in order to check their direct impact in the results.

6.1 Introduction

The main purpose of the Building Management System is two-fold. Concurrently, it tries to minimise energy consumption by avoiding overuse situations and wasteful scenarios, and also tries to deliver the most comfortable scenario for occupants according to their desires. For instance, by predicting the time of entrance of a person in a room, it is possible to adjust the temperature beforehand, or by detecting that a room is empty, lights can be turned *off* if they have been left *on* by mistake. However, there exists a trade-off between such two metrics, and it is needed to arrive to a midpoint in order to maintain people comfortable whilst guaranteeing an acceptable level of energy consumption. It is also possible to create patterns for extreme actuation to favour one of the two metrics, and, for instance, maintain the people at the maximum possible level of comfort without taking into consideration the amount of energy overused.

Section 6.2 presents an overview of the three main stages that data must overcome in order to be generated, transformed, stored and consumed by the different architecture layers previously explained. Then, Section 6.3 explains more in depth the developed BMS responsible for obtaining energy consumption and comfort levels. Finally, Section 6.5 presents the overall conclusions extracted from the execution of the BMS during work days in a specific building.

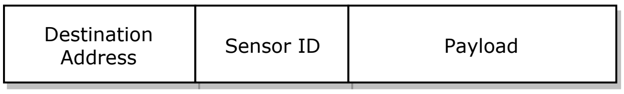
6.2 Data Stages

The following sections explain the three main stages that data must overcome in order to extract valid information from it. First section explains how the data is generated and which elements are being monitored inside the building. Since the deployment of many sensors inside a building is costly, some of them are simulated in order to scale the system and create a more realistic scenario. Then, how this data is transformed into a standard format and later uploaded to the Cloud platform is explained. Then, the consumption of the data by means of the building management application is described.

6.2.1 Data Generation

The first step towards the enhancement of a building with smart features is the monitoring of all the necessary elements inside it. In the case of a building, important elements are lights, HVAC systems, computers, doors and windows. Moreover, the environment needs to also be monitored to know whether we can take advantage of it. For instance, lights can be turned *off* if outdoor luminosity is high enough for indoor working.

Some of the aforementioned elements are endowed with small sensors capable of acquiring the necessary data to deduce their state. In the case of lights, HVAC systems and computers, potentiometers are used to read the amount of energy being consumed and, thus, know their state. Alternatively, doors and windows make use of electromagnetic sensors to know whether they are opened or closed. In the case of environmental data, temperature, humidity and luminosity sensors allow us to exactly read the respective values. For mimicking the rest of the elements of the system, data is generated by means of a software capable of creating the exact same packets as the physical ones.



Destination Address	Sensor ID	Payload

FIGURE 6.1: Data packet abstraction.

Even though the data generation may vary from one type of sensor to another, the packet containing the data and the corresponding headers for a transmission are equal. Figure 6.1 shows the abstraction of the structure of such packets. As can be seen, it is merely formed by the destination address, the sensor identifier and the payload containing sensor readings.

Data generation rate varies depending on the device under monitoring. In the case of environmental conditions and power usage, samples are taken every 5 minutes. Regarding doors and windows, the sample rate remains the same but additionally, if a change in their state is detected, a message is also generated.

Once the sensor data is read and encapsulated in a packet with the shape seen in Figure 6.1, it is transmitted to the closest gateway in each case. Gateways receive the messages by means of different protocols such as ZigBee and Bluetooth Low Energy. Since gateways are enhanced with Internet connection, raw messages are directly forwarded to a central server responsible for the data aggregation and standardisation.

As can be seen, the architecture is capable of combining the usage of real sensors with software defined sensors, which allows for easy scalability and also fast adaptation in the event of testing distinct scenarios.

6.2.2 Data Transformation and Storage

When the messages reach the central server, it firstly reads the sensor identifier to know the type of message contained inside the payload. Once the type has been detected, the message is transformed into JSON standard format with a structure of *"key":"value"*. Additionally, each message contains a time stamp to know when the value has been generated.

The standardised data is then pushed using the REST API to the Cloud service explained in Section 3.2.4. In this Cloud platform, each physical sensor corresponds to a virtual sensor. That means, each physical identifier is assigned to a virtual identifier. This relationship is privately stored inside the central server in order to be able to correctly push future messages to the corresponding virtual sensor. However, virtual identifiers along with sensor information such as model, type of sensor and location is publicly available thanks to an additional Cloud database that stores this data. By doing so, external entities can take advantage of the platform and query specific sensors without the necessity of deploying their own ones.

As it has been mentioned before in Section 3.2.4, not all the sensors registered into the system are publicly available. The necessity to privatise some sensors is directly related to the security and privacy of the users under the monitored environment. For instance, if proximity and movement sensors are publicly available, third persons would be able to know whether the room containing the sensors is empty or not, and take advantage of such information for social hacking.

In order to avoid this, the only sensors that are shared correspond to environmental monitoring such as temperature, humidity and light. For the rest of the sensors, a password is needed to receive the updated values.

6.2.3 Data Consumption

The application developed is composed by two differentiated elements. Firstly, the BMS is responsible for directly receiving sensor information via the subscriptions performed to the different sensors inside the building. By using a lightweight publish/subscribe client, once a new message is stored in the middleware database, it is also forwarded to the application, allowing the BMS to act accordingly if necessary. However, as it has been previously said, since the deployment and testing of such scenario in a real environment is too costly, simulation has been chosen as the second element for acquiring close-to-real results of the benefits of the building enhanced with smart capabilities.

The behaviour of the whole system is as follows. The architecture developed feeds the BMS with both real and software generated sensor data. Once this data reaches the application, simulated elements modify their state in order to be synchronised with the corresponding sensor. For instance, if a sensor from a specific location tells that the lights are *off*, the simulated element must also be *off*. By using this pattern, the simulator maintains the synchronism between the real and virtual sensors with the building simulated elements. Consequently, if the simulator detects that an actuation must be performed, it automatically changes the state of the element and updates all the required software defined sensors in order to maintain the synchronism.

In addition to the simulation of the building elements, people inside it are also simulated to be able to repeat the tests multiple times. People are defined by a set of actions that can be performed inside the building along with the probability over time of this actions to actually be performed. For instance, if the building under simulation corresponds to an office, people are more susceptible to perform the action *enter* during the initial morning hours. The combination of such definitions is stored as the *profile* of the user, allowing different user profiles across the simulated people.

Last feature of the simulator corresponds to the smart capabilities of the BMS. That is, the system must be able to detect whether an action that increases comfort and possibly reduces energy consumption can be carried on by looking at the state of the building at each moment. The implementation is developed by means of a rule-based system that monitors conditions corresponding to every possible actuation to activate. For instance, to know whether the light of a room can be switched *on* or *off* directly depends on the presence of people inside the room, the current indoor light state, outdoor luminosity and windows position.

The comparison between the results of a building enhanced with the aforementioned smart features and a normal one allows to check whether it is possible to increase comfort of the occupants while maintaining acceptable energy consumption levels. Moreover, the flexibility to modify the optimisation criteria creates diverse scenarios in which further actions can be carried regarding energy or comfort optimisation. The calculation of the occupants comfort, however, presents a difficult problem due to the lack of metrics. To this end, we extend literature formulas to include all the sensor information present in the system.

6.3 Smart Building Resource Manager

This section presents the specific case of a smart building resource manager capable of automatically adapt the state of the building in order to achieve an acceptable level of energy consumption whilst maintaining good comfort levels for the occupants, due to the inner trade-off that exists between such two metrics. The optimal situation would be the maximum comfort of the people whilst maintaining the least possible consumption level. However, such scenario is usually utopian: in order to increase the comfort of the occupants, it is commonly needed to interact with the building elements beforehand in order to recreate their desired status for their arrival. Thus, the usage of building elements is increased and, then, the overall building energy consumption.

Following sections define the problem and present the models and elements that contribute to the building management system. Particularly, environmental models such as temperature, humidity and light are introduced. Moreover, the actions that the occupants can perform, and their probability, are also modeled in order to mimic their usual pathing inside the building.

6.3.1 Problem Statement

The *operational phase* of a smart building resource manager is responsible for monitoring and collecting all the necessary building information to deliver correct responses to different scenario situations and interact with the indoor elements to guarantee acceptable levels of comfort and energy consumption.

Indeed, besides energy consumption, another important metric to take into consideration in the operational phase of building automation is the comfort of occupants. In other terms, the environmental conditions in the building should be comfortable and pleasant for all occupants, matching as much as possible their desires. It is clear that pursuing both maximum comfort and minimum energy consumption may lead to contradictory actuations. For this very reason, we take into account both metrics in the design of our BMS and its decision-making process.

The proposed BMS receives two different inputs: i) the environmental conditions such as temperature, luminosity, air quality, etc. and ii) the movements and behaviours of the occupants of the building. With the latter, the BMS sets up and maintains occupants behavioural models. These models contain a set of actions that occupants are likely to perform during the day. For instance, one action included in a model could be that a specific occupant is usually entering in a given room between 7.30 and 8.30 am with a 90% probability. Movements and actions of the occupants can be tracked using both indoor and outdoor location services like in [87].

Using these models, the BMS monitors the status of the different rooms and, according to the expected movements of occupants, operates the actuators in order to minimise energy consumption while maintaining acceptable levels of comfort. In order to provide to the system the time to take such decisions, we introduce a *prediction threshold* in Section 6.3.2, so that the BMS can act in advance with respect to actual occupant actions. For instance, the BMS can turn the HVAC *on* in a given room so that the desired temperature is reached just before the expected arrival of the occupant. Or, if the occupants of a room are expected to go for lunch at 12.30 am for an hour, the BMS automatically turns *off* the lights in the room, places smart plugs in sleep mode, reduces the HVAC use, etc. just after the expected action is performed, without the need of human intervention. As mentioned above, all BMS actuations are taken by also considering the comfort of the occupants in such a way that an acceptable level is always guaranteed.

6.3.2 System Definition

As previously stated, it is important to clearly define the building under study in order to understand the results. This section presents the models and variables utilised in the simulations. Regarding specific building parameters, their values are defined in Section 6.4.

Occupants can either be employees sharing rooms, area professors with their own offices, or external personnel like visitors, building caretakers, and janitors. Clusters of three different levels are considered: room, zone (a given set of rooms), and floor. Regarding the type of sensors that can be placed in a room, we consider (i) a single sensor able to gauge temperature and luminosity and to detect whether the room is empty and (ii) a sensor that detects the status of a computer. In zone clusters, humidity and gas detection sensors can be installed. Only one alarm sensor is placed in the floor cluster. Concerning the actuators, each room is supposed to contain HVAC, smart plugs, and actuators for windows and doors.

Environment

Environmental conditions have an important impact on the energy consumption of a building. Specifically, temperature, humidity, and luminosity are considered for our study. A model is provided for each of them. Table 6.1 shows the average, minimum and maximum values considered for these three parameters, empirically collected on a winter day in Barcelona. The difference between these values and the conditions desired by building occupants mandates the use of indoor HVAC and lights.

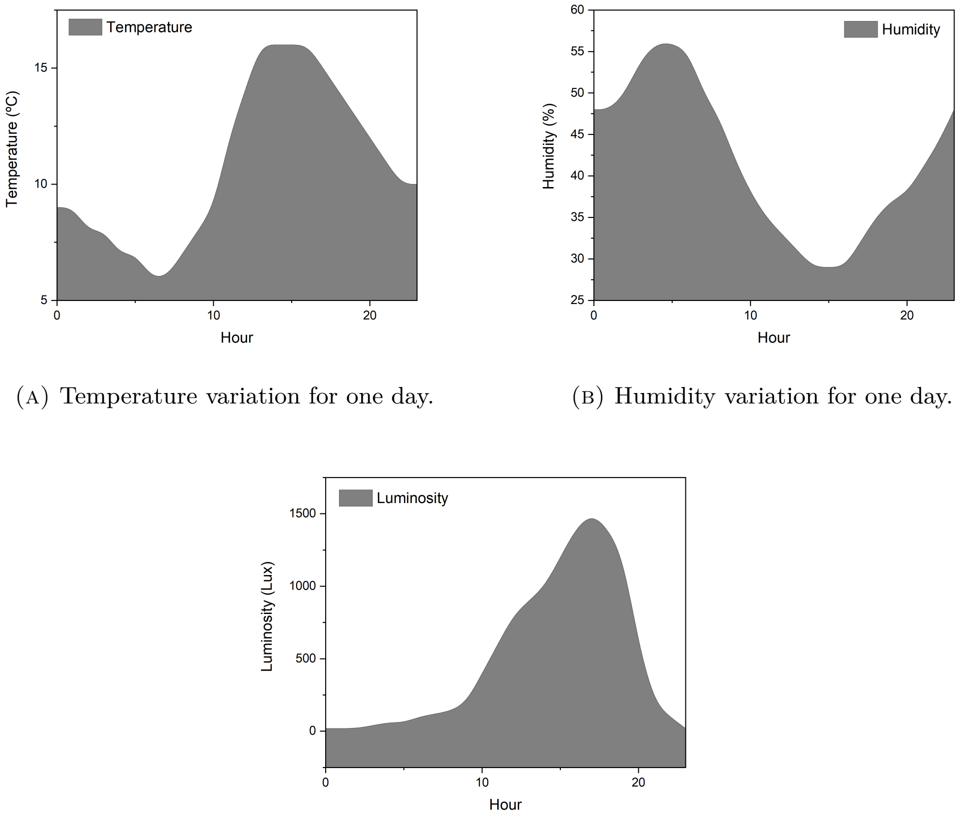
Parameter	Average	Min	Max
Temperature (°C)	10.8	6	16
Humidity (%)	41.87	29	56
Luminosity (lx)	505	20	1,500

Table 6.1: Average, minimum and maximum environmental condition values.

Additionally, Figure 6.2 shows the variation of the temperature, humidity and luminosity parameters during the day. As it can be observed, temperature always lies below the desired level of 21 °C. During the morning, environmental temperature increases from its lowest value, and reaches the maximum at midday. During the afternoon, temperature starts to decrease and arrives to its minimum, again, at night. Similarly, luminosity starts to increase as the sun rises, and starts to decrease at afternoon, until it reaches its lowest value at night. Humidity, however, follows the opposite tendency. During morning and night it stays high, whilst in midday it reaches its lowest values.

Occupants

The occupants of a building are an important external agent that needs careful modelling. Occupants are categorised into different profiles, each of which contains a set of actions that can be performed. Every action is associated to the probability of its occurrence, specified



(A) Temperature variation for one day.

(B) Humidity variation for one day.

(C) Luminosity variation for one day.

FIGURE 6.2: Temperature, humidity and luminosity variations for one day.

hourly across a full day. That is, a user profile U contains the set of actions $a_1, a_2, a_3, \ldots, a_n$, where each action a_i is represented as a set $a_i = \{p_1, p_2, \ldots, p_{24}\}$ of 24 values $0 \leq p_i \leq 1$, one for each hour.

The set of possible actions that occupants can execute strongly depends on the type of building and how occupants behave whilst inside. For our study, the considered actions to perform are $\{enter, move, meeting, lunch, exit\}$. Figure 6.3 shows the available location transitions associated to these actions.

As it can be seen, the first action always needs to be *enter*. Once inside the building, any occupant can choose to *move* to a specific room, go for *lunch* or participate in a *meeting*. To *exit* the building, it is needed for the occupant to either be in a room or in a meeting. Exiting the building is also a requirement in order to terminate the simulation, as the building has to be empty at the end of the day.

The probability with which every occupant can decide to execute an action varies throughout the day and depends on the role of the occupant. Occupants are divided into three different roles: students, area professors, and external personnel. The action probabilities that define each of such different roles are depicted in Figures 6.4, 6.5 and 6.6.

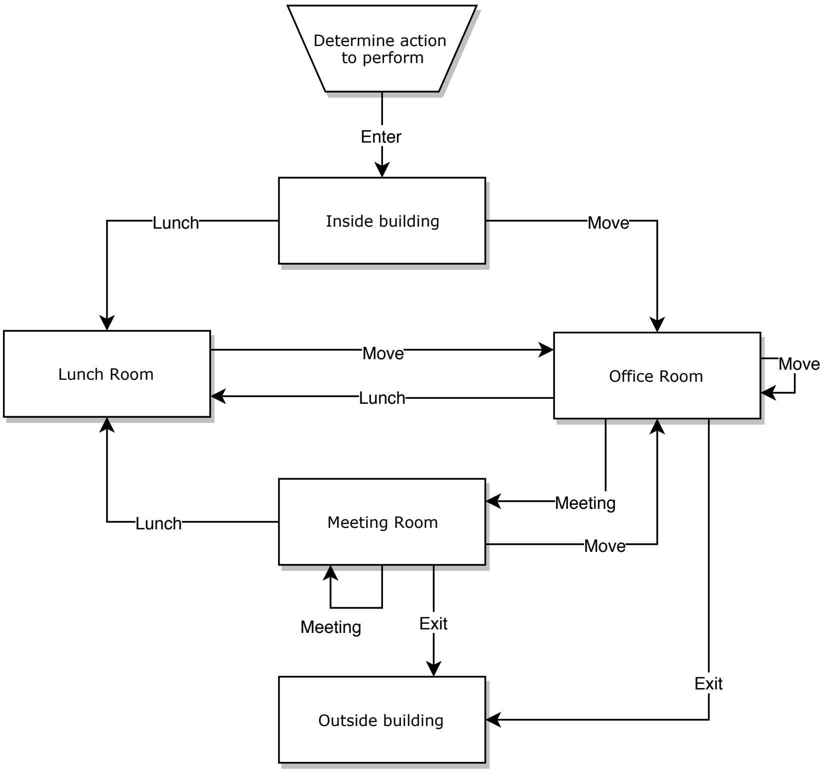


FIGURE 6.3: Location transition graph based on the current defined actions.

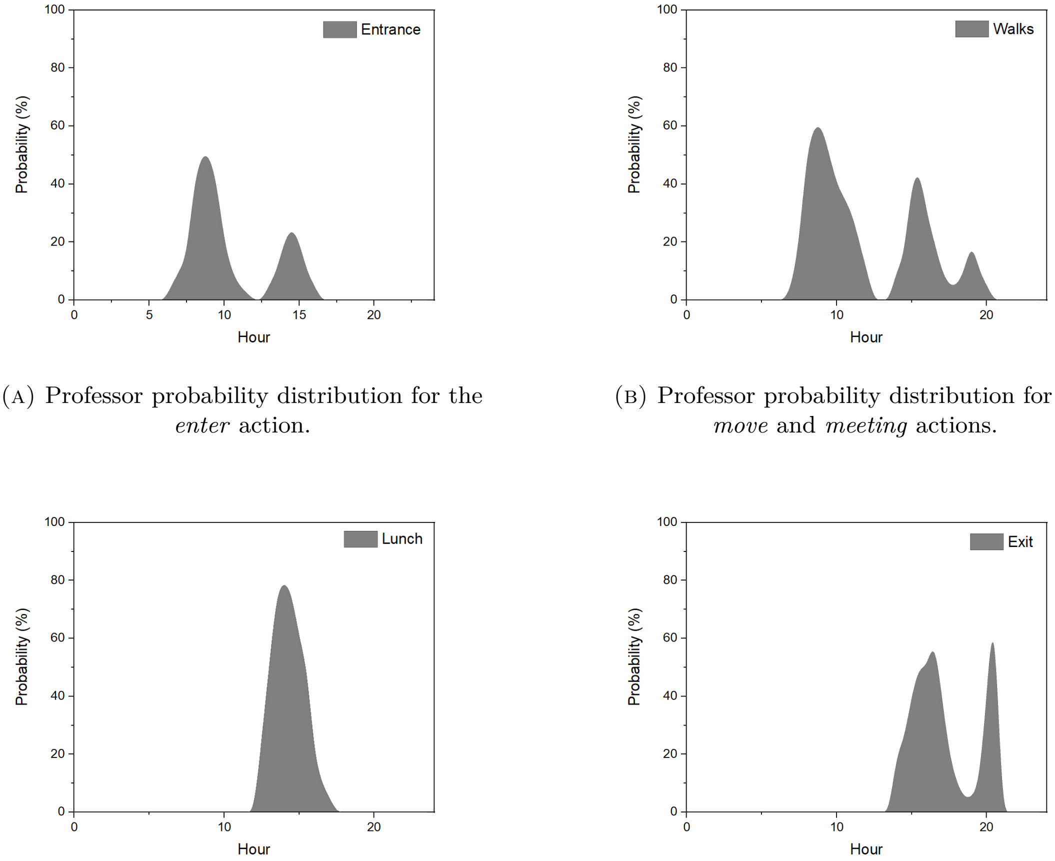
Figure 6.4 shows the probabilities over time for each of the possible actions that can be executed by professors. As can be seen in Figure 6.4a, the probability of entering the building is maintained relatively high during the majority of morning hours. This is due to the fact that entrance is not usually imposed, and some professors might have previous activities that delay their entrance. After lunch, the probability increases again due to potential professors working on afternoon classes.

Regarding their movement, Figure 6.4b shows that, during the morning, it is highly possible for a professor to leave his office. This is mainly due to their lectures or due to meetings in shared rooms. Such trend is also maintained after lunch, but with lower probability. These movements can last from 5 minutes up to 120 minutes, depending on its purpose.

Lunch is usually centred during the midday, but, as can be seen in Figure 6.4c, some professors might need to have lunch very early to avoid queues and wasted time. Regarding the duration of their lunch time, it can vary between 30 minutes and 90 minutes.

The exit of the building, shown in Figure 6.4d, is mainly centred after lunch and during the afternoon. Most professors have a complete working schedule and their activities do not allow them to leave their office before the evening.

Figure 6.5 shows the probabilities over time for students. As can be seen, the flexibility of such actions is limited, since students usually have common fixed schedules due to their lecture hours. Regarding their entrance, as shown in Figure 6.5a, it usually happens during a tight morning time interval (between 7 am and 9.30 am), or after lunch, for the students with



(A) Professor probability distribution for the *enter* action.

(B) Professor probability distribution for the *move* and *meeting* actions.

(C) Professor probability distribution for the *lunch* action.

(D) Professor probability distribution for the *exit* action.

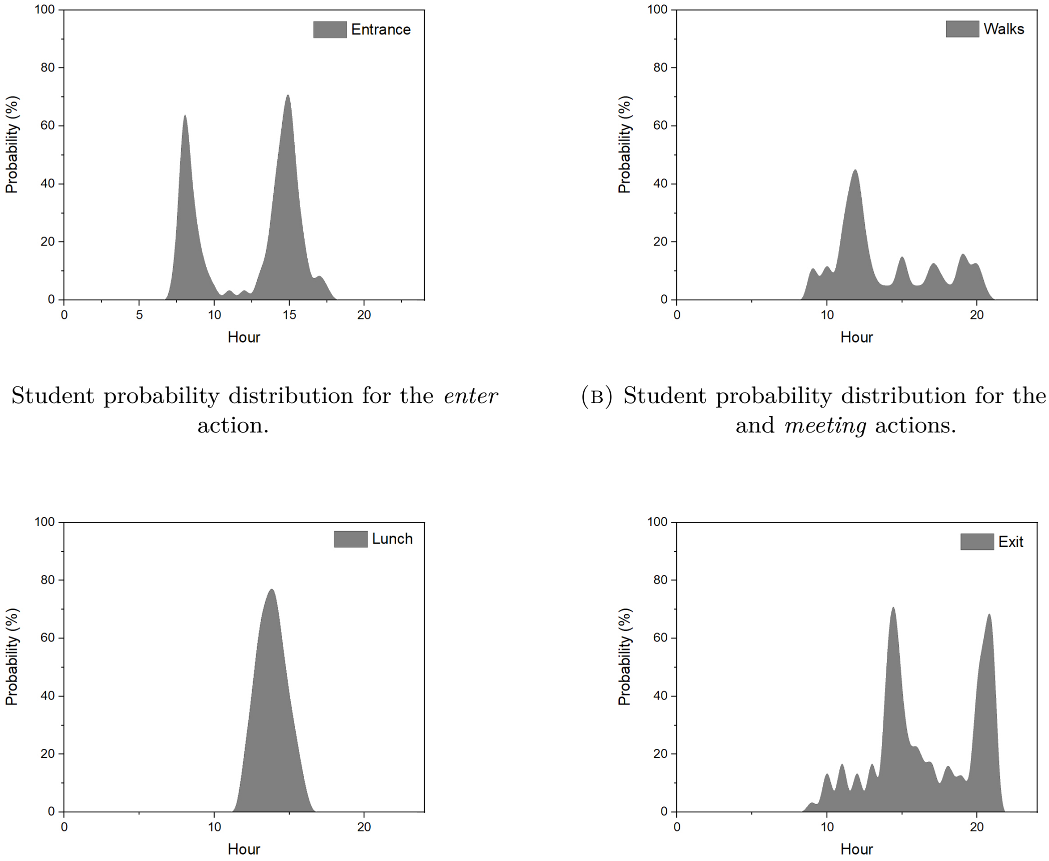
FIGURE 6.4: Professor user profile probability distributions.

afternoon timetables. This second interval is wider since afternoon entering hours may vary between 1 pm and 4 pm.

With respect to their movements, shown in Figure 6.5b, students have a more relaxed behaviour, since they tend to leave classrooms once in a while. Even though the probability exists almost throughout the whole working day, it can be seen that, during midday, the probability increases. The duration of these movements is significantly small, from 5 minutes to 20 minutes, since lecture hours are always one after the other. Lunch probabilities (Figure 6.5c) are very similar to professor ones, since they both share the same dinning places and most of the working hours. Students have between 30 minutes and 60 minutes to have lunch.

Figure 6.5d shows the exit probabilities, which are the most different ones compared to other user profiles. Even though the majority of the students share equal timetables, few of them have reduced ones. This behaviour is expressed by having significant exit probability after every lecture class. Apart from that, the two common exit intervals are present (2 pm to 3.30 pm and 8 pm to 9.30 pm).

As can be seen in Figure 6.6, the time intervals in which PAS personnel execute actions



(A) Student probability distribution for the *enter* action.

(B) Student probability distribution for the *move* and *meeting* actions.

(C) Student probability distribution for the *lunch* action.

(D) Student probability distribution for the *exit* action.

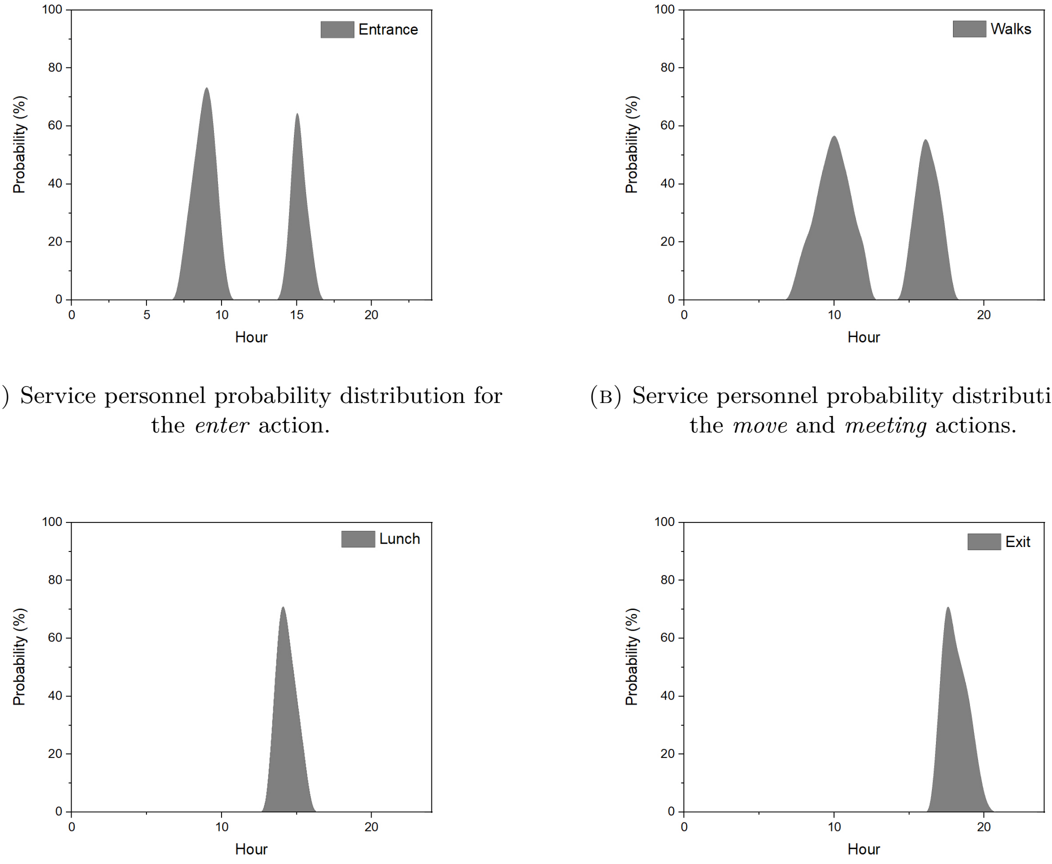
FIGURE 6.5: Student user profile probability distributions.

are very sharp due to the fact that they usually have a fixed schedule with little flexibility. Current entrance schedule, depicted in Figure 6.6a, starts at 7.30 pm and lasts for 3 hours until the afternoon turn enters at 3 pm. This is a common schedule that can vary in few cases.

PAS people group multiple sectors, from administrative roles to cleaning services. This wide role coverage makes it difficult to accurately mimic their movements, since a portion of PAS people are always inside their office, but others tend to move repeatedly. This has been solved by providing a relatively high movement probability for most of the working hours, as can be seen in Figure 6.6b. This heterogeneity is also shown in the duration of such movements, which can vary between 5 minutes and 60 minutes.

Lunch time, shown in Figure 6.6c, is fixed between 1.30 pm and 3.30 pm depending on their turn, since many PAS working places cannot be unattended for a long period of time. PAS people have between 30 and 60 minutes to have lunch.

As regard to the exit (Figure 6.6d), their schedule varies from 4.30 pm up to 7.30 pm. This wide range is due to the heterogeneity of PAS roles and the variability in the number of contracted hours.



(A) Service personnel probability distribution for the *enter* action.

(B) Service personnel probability distribution for the *move* and *meeting* actions.

(C) Service personnel probability distribution for the *lunch* action.

(D) Service personnel probability distribution for the *exit* action.

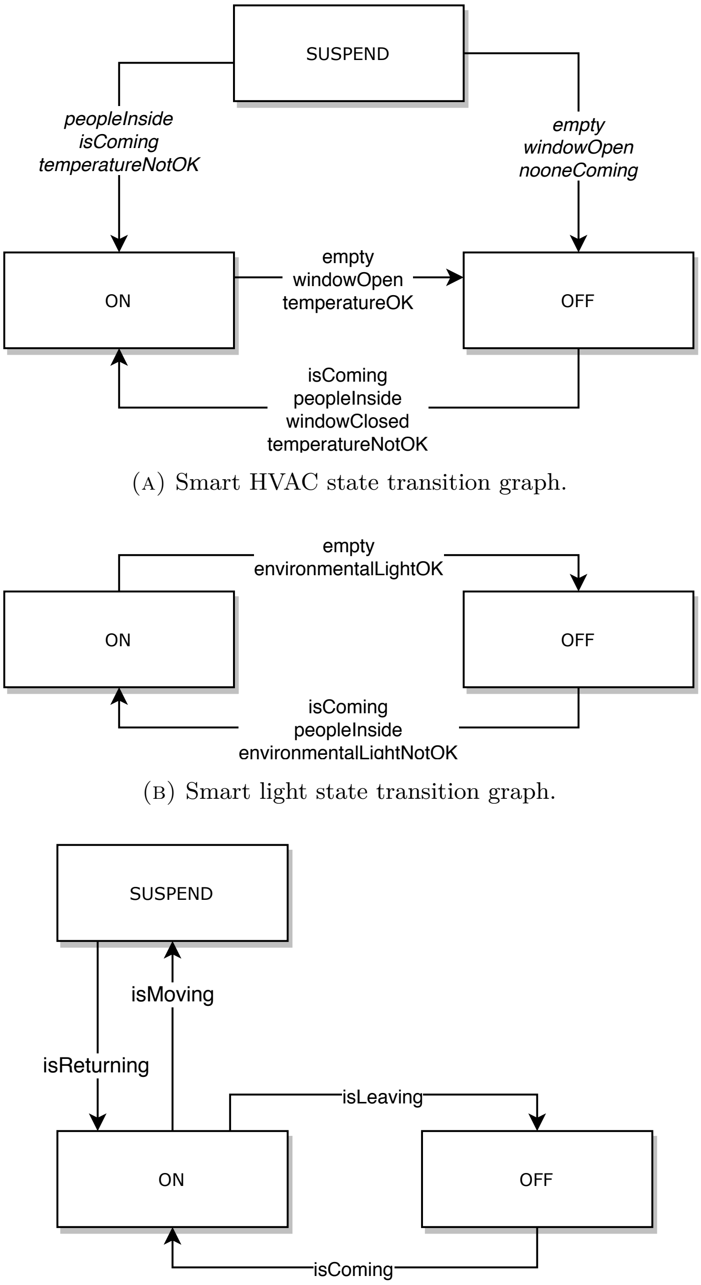
FIGURE 6.6: Service personnel profile probability distributions.

Each occupant user profile also indicates the desired environmental conditions for reaching the most comfortable scenario. Specifically, values for temperature and luminosity are set. Currently, temperature and luminosity are set to 21 °C and 600 lux, respectively, and the system aims for delivering such conditions once a person is expected to enter any room. In the event of a room shared by many people, such parameters are calculated as the mean of the desired values of the different occupants, in order to deliver a reasonable comfort level to all of them.

It is worth mentioning that, currently, these occupants behavioural models have not been constructed using real data. This problem has been left for future works. On the contrary, in this thesis, we are interested in testing whether taking smarter and automatic decisions in the BMS using these models effectively provides an improvement both in energy consumption and occupants' comfort.

Smart actuation modes

Our solution is able to launch automatic actuations for the different elements of the building in order to change their state, without the necessity of human intervention.



(A) Smart HVAC state transition graph.

(B) Smart light state transition graph.

(C) Smart computer state transition graph.

FIGURE 6.7: State transition graphs for the monitored elements.

Figure 6.7 shows the state transition graphs for the different elements of the building, namely the HVAC, the lights and the smart plugs connected to the computers. Computers and HVAC can either be in *off*, *suspended* or *on* state while the lights can be in *off* or *on* state. Each arrow is labeled with the combination of conditions required for executing the corresponding state transition. Specifically, we define the following conditions:

People *peopleInside* or *empty*, to detect whether there are people inside a room or it is empty.

Windows *WindowOpen* or *WindowClosed*, to detect if a window is either open or closed.

Temperature *TemperatureOK* or *TemperatureNotOK*, to detect if the temperature is the one desired by occupants or not.

Luminosity *environmentalLightOK*, to detect if the luminosity of the environment is enough and no artificial light is needed.

Entrance/Exit *isComing* or *isLeaving*, to predict that someone is going to either enter in a room for the first time or leave until the next day.

Movement *isMoving* or *isReturning*, to predict that someone is either moving outside or returning to a room from a meeting or lunch.

The last two conditions depend on the actions of the occupants. With the models previously defined, the BMS is capable of predicting such actions. To give time to the BMS to react and prepare the building for the occupants, we define a *prediction threshold*. This threshold allows the BMS to act in advance with respect to the expected future actions, but only up to an extent. The value of this threshold clearly impacts both comfort and energy consumption; for this very reason, we perform an evaluation of such impacts in Section 6.4 in order to find the optimal trade-off.

Therefore, if, for instance, an occupant is going for lunch (*isMoving*), his computer is *suspended*; and it is brought back to *on* if the occupant is expected to return (*isReturning*) soon. We consider that the transition between the states of a computer requires 5 minutes. Similarly, if, for instance, a room is empty, its light is switched *off*; but, if an occupant is expected to arrive (*isComing*) and the environmental luminosity is not OK, the light is turned *on*. In this case, the state transition is instantaneous.

The behaviour of the HVAC is slightly different, as the temperature cannot be adjusted instantaneously. To take this latency into account, we consider three parameters, namely the desired, the environmental, and the current temperature. The former is the temperature desired by the occupants of a room. When the HVAC of a room is *on*, the temperature gradually reaches the mean of the temperatures desired by the people inside; conversely, when the HVAC is either *suspended* or *off*, the temperature tends to the environmental one. We use Equation 6.1 to update the temperature of a room. Every simulation step (10 seconds), a fraction of the absolute difference between the current and the desired temperature is either subtracted or added to the current temperature. The F factor varies depending on the state of the HVAC. If the HVAC is *on*, the temperature varies at a $F = 1\%$ rate until the desired temperature is reached. If it is in either *suspended* or *off* state, the temperature changes at a $F = 0.25\%$ or $F = 0.5\%$ rate, respectively, until the environmental temperature is reached.

$$T_{new} = T_{current} \pm |T_{\{desired/environmental\}} - T_{current}| * F \tag{6.1}$$

6.3.3 Metrics

The two main metrics considered in our work are the overall building energy consumption and the comfort of the occupants. Following sections aim at defining such metrics in more detail.

Energy consumption

We consider that the building energy consumption depends on the HVAC systems, lights and computers. The energy consumed by the WSN may also be taken into account. The WSN is, however, considered fixed and immutable, and, therefore, its consumption is constant and it has been neglected for this part. Table 6.2 shows the power consumed by the three elements depending on their states. Thus, the overall energy consumption is calculated according to the state of these elements and their power consumption during the entire simulation time (1 day).

Device	OFF	Suspended	ON
HVAC	0	800	1,500
Lights	0	-	200
Computer	0	50	350

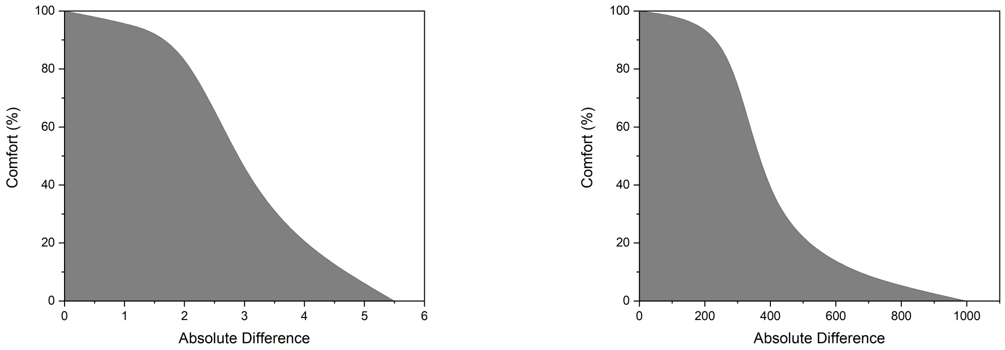
TABLE 6.2: Energy (in kWh) consumed by the monitored devices in the different possible states.

Comfort

The definition of comfort has been widely studied in the literature and many different formulations have been proposed, as commented in Section 2. Many authors agree to consider thermal comfort as the most important parameter. However, since we are pursuing the automation of some elements in a smart building according to the expected actions of the occupants, we also include the adequacy of this automation in our formulation. Therefore, in our work, the definition of comfort aggregates several values with their proper weighting. In particular, thermal, light and *device readiness* comforts are considered.

The temperature comfort makes reference to the absolute difference between the desired temperature and the current one. The desired temperature is set to 21 °C and the system aims for delivering such condition once a person is expected to enter any room. In the event of a room shared by many people, the parameters are calculated as the mean of the desired values of the different occupants, in order to deliver a reasonable comfort level to all of them. Figure 6.8a shows the model used for assessing thermal comfort. It is worth noticing that, when the desired and the current values widely differ (4-6 °C), comfort is very low (lower than 20%). However, small differences (less than or about 1 °C) are better tolerated, with comfort close to 100%. Reaching the maximum comfort, thus, may require substantial additional energy consumption and might be unnecessary.

Similarly, light comfort stands for the absolute difference between the desired and current room luminosity. Figure 6.8b shows the utilised luminosity model. The desired luminosity is set to 600 *lux* and, again, the system aims for delivering such value once a person is expected to enter any room. As for the case of the temperature, the luminosity model indicates a comfort close to 100% if the difference between the desired and the current value is small (less than or about 200 *lux*) and becomes very low (comfort close to 0) if the difference exceeds 400 *lux*.



(A) Thermal comfort percentage depending on absolute temperature difference.

(B) Light comfort percentage depending on absolute luminosity difference.

FIGURE 6.8: Comfort percentages for the absolute difference between current and aim values.

Finally, the device readiness comfort is related to the prompt availability of the users' devices inside a room. For instance, computers must be fully operational when an occupant enters a room, instead of being *off* and needing to be turned *on* manually. For this case, no figure is shown since device readiness comfort follows a boolean tendency: when the computer is switched *on* and ready for use, the comfort is maximum, whereas it is 0 in all other cases (*off*, *suspended* or transitioning from one state to another).

$$C = 0.57 * C_{luminosity} + 0.38 * C_{thermal} + 0.05 * C_{device} \tag{6.2}$$

The overall occupants' comfort formulation is provided in Equation 6.2. The weights used for the thermal and luminosity comforts are similar to those proposed in [88]. However, we have added the device readiness comfort, assigning a weight of 5% to it in order to preserve the importance of both temperature and luminosity.

Building

The building under study represents an important aspect inside the management system. Buildings are commonly divided into rooms, each of which contains a series of devices that occupants interact with. For our study, the building is divided into rooms of three different types, namely office, meeting and lecture rooms. Office rooms are individual places that professor utilise whilst they are not in a meeting or imparting a lecture; meeting rooms are meant to be shared places for holding reunions or talks; lecture rooms are bigger places in which professors impart their lectures to a group of students.

Table 6.3 shows the different types of rooms that the building contains. Concretely, for each type type of room, the number of HVACs, lights and computers is determined. These

Room	Description	HVACs	Lights	Computers
Office	Room for professors	1	1	3
Meeting	Room for hosting events, talks, reunions, etc.	1	1	2
Class	Room imparting computer classes to students	1	1	20

Table 6.3: Description of the distinct types of rooms and the amount of monitoring elements each contains.

elements are automatically managed via actuations by the management system, and no interaction is needed by the occupants.

As can be seen, HVAC and lights are single elements present in every room and utilised by all the occupants simultaneously. On the contrary, computers are defined as individual components that can only be utilised by one person at a time. In the event of a student entering a room with no available computer, he has to wait until one becomes available.

6.3.4 Operation

This section presents the abstract implementation of the operational phase of the Building Energy Management System. That is, the phase in which the BEMS utilises the data from the aforementioned models and sensors to evolve, interact and actuate upon the elements of the building to aim for the optimal scenario under the current considered metrics.

During this phase, the tool monitorises energy consumption and comfort of each occupant. Once simulations finish, it is then possible to study and understand the trade-off that exists between those two metrics depending on the prediction threshold used.

Algorithm 5 shows the entry point of the simulations. The input of the algorithm is represented by a set of rooms with all the required information already defined, such as the room type, number of sensors inside each one or the type of the sensors.

The repeatability of the simulations is guaranteed by storing all the necessary information in independent databases. Thereby, a set of databases is initialised prior to running the simulations. Concretely, there exists the following databases:

Building Stores the structure of the building in rooms. Each room is defined as the number of sensors per type as well as the number of entities present. An entity can be a door, window, computer, HVAC, lamp, etc.

People Stores the name and the profile of each person capable of entering the building during the simulation.

Events After running the simulation, all the movements performed by the people are stored in order to exactly mimic such behaviour in the future.

Comfort Stores the comfort evolution of each occupant.

Algorithm 5 runBuildingManagementSystem(*rooms*)

```
Require:
  rooms: set of pairs with number of rooms per type
Ensure:
  $Databases \leftarrow \emptyset$
  $Building \leftarrow \{\}$
  $People \leftarrow \emptyset$
$Databases \leftarrow$ InitialiseDBs($settings$)
$Building \leftarrow$ ObtainBuilding($Databases.BuildingDatabase, rooms$)
$People \leftarrow$ ObtainPeople($Databases.PeopleDatabase$)
$Manager \leftarrow$ CreateManager($Building, People$)
switch ($SimulationMode$)
case 1:
  BaseSimulation()
case 2:
  SmartSimulation()
case 3:
  RepeatSimulation()
default:
  $IncorrectMode$
end switch
```

Thus, prior to starting the simulation, the aforementioned databases are initialised and the required parameters are read. Specifically, the building and people data are read from their respective databases. Then, the manager or controller is initialised, which is responsible for the decision-making process as well as for the metric monitoring. Specifically, energy consumption and occupants' comfort.

Currently, the manager accepts two separate execution scenarios, namely *base* and *smart*. The *base* scenario tries to mimic as close as possible the behaviour of building elements when no automation is present. In particular, the following behaviours for building elements are considered:

HVAC System Maintained *on* during the whole working period of the day, which lasts from 7 am to 7 pm. The current HVAC behaviour equals to our assumption since the centralised system is not capable of switching *off* several parts of the system.

Lights Kept *on* while there are someone inside the room, switched *off* with a 50% probability when occupants leave the room (for meeting or lunch), and switched *off* at the end of the working day. Since no automatic actuations are currently present, many rooms remain with the lights switched *on* due to short departures or occupant mistakes.

Computers Switched *on* when its user enters the room, switched *off* with a probability of 50% when the user leaves the room, and switched *off* at the end of the working day. Currently, many occupants leave their computer *on* the whole day, even when they are not in the building anymore.

As for the *smart* mode, actuations and element behaviour is explained in Section 6.3.2.

Algorithm 6 Simulate($Manager, Building$)

```
Require:
  Manager: controller for managing the data
  Building: structure of the building
Ensure:
  $Current\_Step = 0$
while $Current\_Step < MAX\_STEPS$ do
  $Manager$.UpdateActions($Manager.People$)
  $Manager$.UpdateComforts()
  $Building$.FireRules()
  $Building$.UpdateConsumption()
  $Current\_Step = Current\_Step + 1$
end while
```

Algorithm 6 shows the actual simulation steps independently of the simulation mode specified. The simulation considers a full day of 24 hours. The simulation is divided into steps of 10 seconds, meaning that a day is represented by 8,640 steps. At each step, the simulation considers the current building status and verifies if an occupant action is expected in the near future, i.e., within the prediction threshold window introduced in Section 6.3. For instance, if a threshold of 150 steps is considered, the BMS checks whether an action is probable in the next 25 minutes according to the occupant's behavioural models. So, if an occupant is expected to be entering an empty room, the system acts consequently, by switching the HVAC *on* in advance, so that the room is at the desired temperature when the occupant arrives. To do so, the simulator firstly updates the actions of the occupants. Then, comforts are calculated and stored in the aforementioned comfort database. After updating the occupants, it is possible to fire the building's decision-making rules to check whether it is possible to modify the status of indoor elements to optimise the aiming metrics. Finally, the consumption is updated and stored.

Algorithm 7 UpdateActions($People$)

```
Require:
  People: set of people
for all $PersonpinPeople$ do
  if $p$.IsActing() then
    if $p$.HasFinished() then
      AssignAction($p$)
    else
      DecreaseSteps($p$)
      ExecuteAction($p$)
    end if
  else
    AssignAction($p$)
  end if
end for
```

Algorithm 7 shows the final piece of the simulator, which is responsible for updating the actions of the occupants. In particular, it checks whether each person is already performing an action. If so, it checks whether such action is finished in order to assign a new action or update the current one. Note that the assignation of actions is performed by following the probabilistic models presented in Section 6.3.2.

6.4 Evaluation

This section presents the evaluation of the building energy management system with the characteristics and parameters previously explained. Due to the elevated cost of deploying and testing the previous results inside a real scenario, it has been decided to simulate the evolution of the building in order to see whether the monitoring of the indoor elements allows for the implementation of smarter rules to enhance both energy consumption and the comfort of the occupants.

Simulations have been divided into three distinct scenarios to understand and identify which building layouts and number of occupants best benefit from the enhancement of the building with smart capabilities. In particular, three simulations are considered.

Section 6.4.1 shows the results of a building mainly composed of office rooms, with professors as the main source of occupation. Sections 6.4.2 and 6.4.3 depict the results of a study building composed of classrooms and few office rooms. However, the former simulation is performed with a large number of students, whilst the latter contains less students. This differentiation is evaluated to check the impact of occupants under the same building conditions.

Results for the three simulations display a daily energy consumption evolution comparison between the base and the distinct smart prediction threshold values, as well as an overall energy consumption comparison. Mean comfort of the occupants is also depicted in order to check the minimum required smart threshold that exceeds the comfort of the base case. Finally, a success rate comparison is shown. This success rate is related to the status of a room once a person enters. A success is considered when the room is delivered with the desired state predefined by the occupant.

Desired environmental conditions for the occupants are set to 21°C and 600 lux, as previously stated in Section 6.3.2.

6.4.1 Simulation #1: Offices

This simulation is characterised by the specification of a building layout mainly composed of office rooms, with few meeting rooms and class rooms. Table 6.4 defines the parameters utilised for the specification of the building layout and the number of occupants of each type.

As can be seen, occupants are mainly composed of professors, with few students attending lectures, and external personnel for managing purposes.

Parameter	Value
Offices	40
Meeting rooms	5
Classrooms	15
#Professors	150
#Students	60
#External personnel	10

TABLE 6.4: Building structure and number of occupants for Simulation #1.

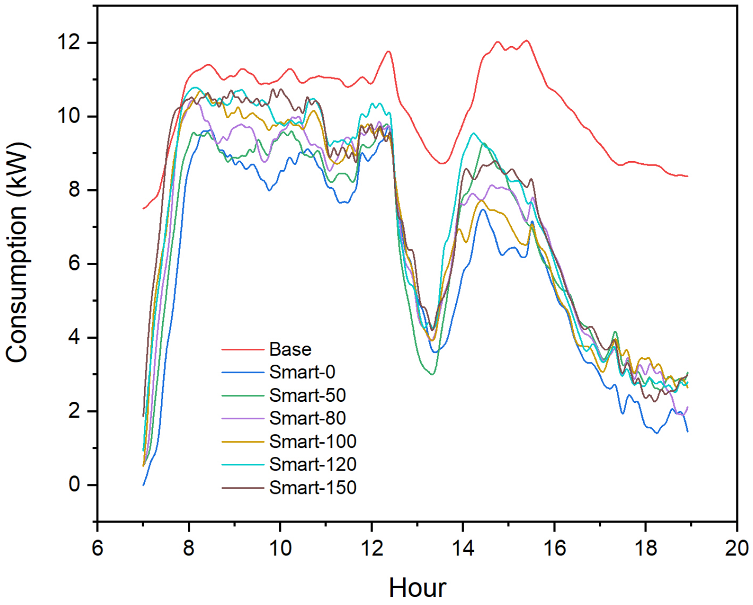


FIGURE 6.9: Daily energy consumption evolution for base and smart threshold scenarios for Simulation #1.

Figure 6.9 depicts the daily energy consumption evolution for the base case and for the distinct smart threshold value cases. Energy consumption values are sampled every 5 minutes and only the working hours, from 7 am to 7 pm, are shown, since the rest of the day the building remains closed and empty. As can be seen, the base case starts with high energy consumption mainly due to the HVAC system switching *on*. Such tendency is maintained until midday, in which rooms are emptied and actuations upon indoor elements can be performed to reduce wasteful situations. When occupants return from lunch hours, energy consumption increases for a short period of time, and slowly starts decreasing until the working hours finish, in which all the system is shut down and energy consumption is negligible.

As for the smart cases, it can be seen that all of them present a similar tendency. Specifically, due to the possibility to individually actuate upon HVAC systems, initial energy consumption remains low, and it keeps increasing until all the occupants are inside the building. During morning working hours, it can be seen that energy consumption varies sharply than the base case, due to the possibility to switch HVACs *off* once a room has been emptied. Such reason also applies to the high decrement during lunch hours. Even though an increment is seen during the afternoon, energy is optimally managed whilst occupants leave, and, at the finish of the work period, few devices contribute to the energy consumed.

The main difference between smart cases lies in the initial energy consumption. As the threshold increases, so does such initial consumption. Since the building is able to predict actions before, actuations upon indoor elements to deliver comfort scenarios can be performed beforehand, and, thus, elements start consuming energy ahead.

Threshold	**Energy Consumption (kW)**	**Mean Comfort (%)**	**Success Rate (%)**
Base	1,479.36	93.8	79.13
Smart 0	857.19	90.59	91.61
Smart 50	953.3	93.21	92.88
Smart 80	993.65	94.28	93.4
Smart 100	1,003.93	95.03	93.68
Smart 120	1,055.23	95.31	94
Smart 150	1,065.42	95.94	94.13

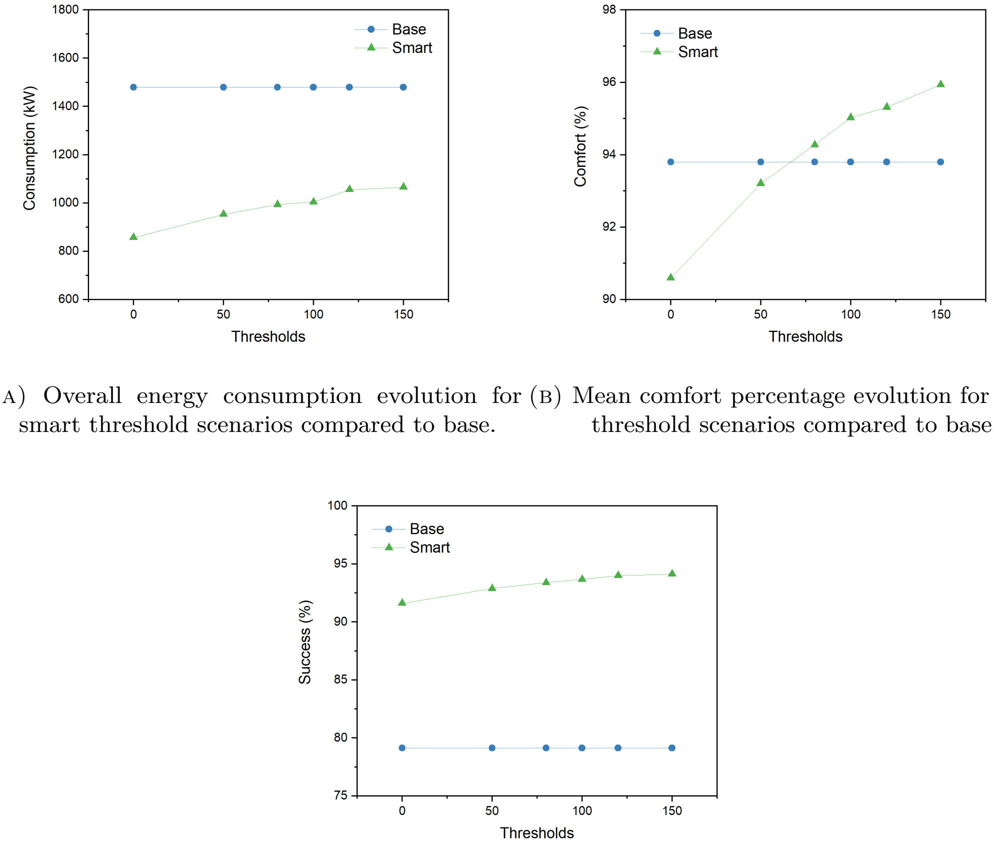
Table 6.5: Overall values for energy consumption, mean occupants' comfort and room success delivery rate for Simulation #1.

Table 6.5 summarises the overall metric values obtained for the different cases for energy consumption, mean occupants' comfort and room success delivery rate. Figure 6.10 depicts such values in order to extract conclusions clearly.

Specifically, Figure 6.10a depicts the overall energy consumption for the base and the smart threshold cases. As can be seen, smart energy consumption increases as the threshold increases and it is always maintained below the base case consumption. In conclusion, the optimisation of indoor elements despite of the threshold utilised always delivers better consumption results.

Similarly, Figure 6.10b details the mean occupant comfort for the base and the smart threshold cases. By utilising a small prediction threshold (below 50), the BEMS is not capable of bringing better comfort than the base case. The base case delivers almost 94% whilst smart threshold below 50 can only reach 93.21%. However, as the threshold increases, the comfort of the occupants surpasses the base case, attaining almost 96% comfort with the maximum prediction threshold.

Finally, Figure 6.10c presents the comparison between the base case success delivery rate and the smart thresholds. Remind that success delivery rate corresponds to the number of times each rooms is set to the desired conditions once the occupant enters. This metric is partially related to the comfort. Whilst the former can be considered as a boolean variable, the latter calculates the actual comfort of the occupant when entering. As can be seen, the success increases as the threshold increases, from 91.61% to a maximum of 94.13% and it always stays above the 79% of the base case. It is worth noticing the big gap (12-15%) between the base and smart cases.



(A) Overall energy consumption evolution for smart threshold scenarios compared to base.

(B) Mean comfort percentage evolution for smart threshold scenarios compared to base.

(C) Room delivery success evolution for smart threshold scenarios compared to base.

FIGURE 6.10: Energy consumption, comfort and success comparison between smart and base scenarios for Simulation #1.

6.4.2 Simulation #2: Crowded Class Building

This simulation is characterised by the specification of a building layout mainly composed of classrooms, with few meeting rooms and office rooms. Table 6.6 defines the parameters utilised for the specification of the building layout and the number of occupants of each type. Moreover, the number of occupants has been increased with respect to the first simulation.

Parameter	Value
Offices	5
Meeting rooms	2
Classrooms	40
#Professors	20
#Students	400
#External personnel	5

TABLE 6.6: Building structure and number of occupants for Simulation #2.

As can be seen, the number of occupants is increased, and students compose the main source of occupancy. Such decision is intended to check whether a crowded building permits little margin of manoeuvre with respect to the base case.

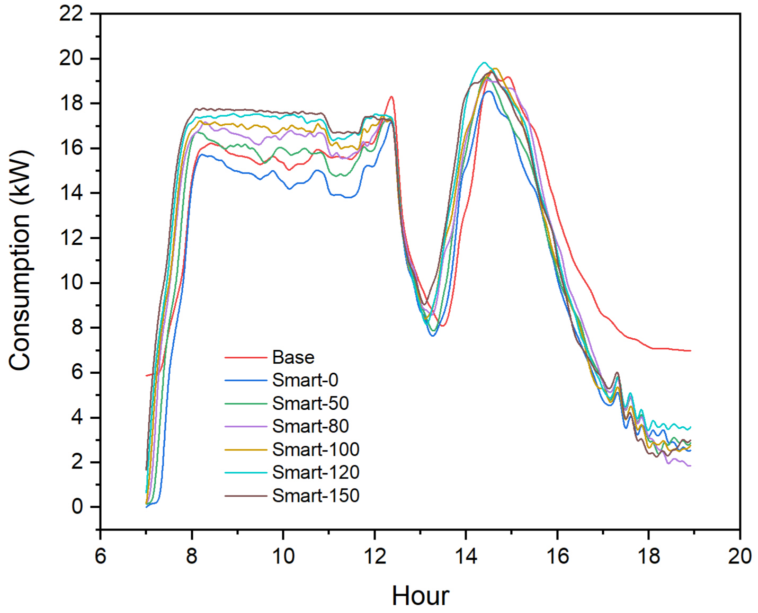


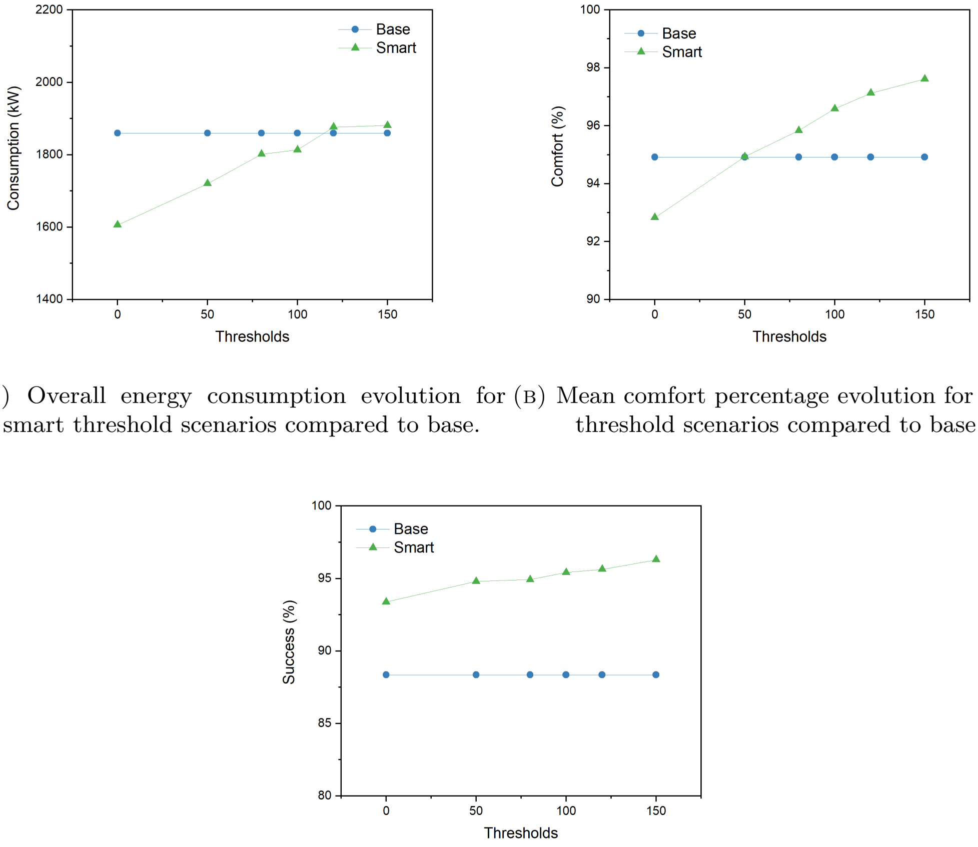
FIGURE 6.11: Daily energy consumption evolution for base and smart threshold scenarios for Simulation #2.

Figure 6.11 depicts the daily energy consumption evolution for the base case and the different smart threshold values. Unlike Simulation #1, variation between the base and the smart cases is almost nonexistent. The utilisation of high number of occupants forces the BEMS to maintain elements always *on*, due to the constant stream of people. That is, there are no time periods in which rooms are empty. Therefore, it can be seen that, during the morning, smart cases behave equally or even worse than the base case. However, at the afternoon, when rooms start getting emptied, smart cases allow energy savings due to the possibility to individually turn room HVACs *off* and the assurance of switching lights and computers *off*.

Threshold	**Energy Consumption (kW)**	**Mean Comfort (%)**	**Success Rate (%)**
Base	1,859.69	94.92	88.35
Smart 0	1,605.84	92.83	93.37
Smart 50	1,720.17	94.93	94.79
Smart 80	1,801.64	95.84	94.92
Smart 100	1,813.36	96.58	95.42
Smart 120	1,876.22	97.13	95.93
Smart 150	1,881.06	97.61	96.29

TABLE 6.7: Energy consumption, comfort and success comparison between smart and base scenarios for Simulation #2.

Equally to Simulation #1, Table 6.7 summarises the overall metric values obtained for the different cases for energy consumption, mean occupants' comfort and room success delivery



(A) Overall energy consumption evolution for smart threshold scenarios compared to base.

(B) Mean comfort percentage evolution for smart threshold scenarios compared to base.

(C) Room delivery success evolution for smart threshold scenarios compared to base.

FIGURE 6.12: Energy consumption, comfort and success comparison between smart and base scenarios for Simulation #2.

rate. Figure 6.12 plots such values in order to extract conclusions clearly.

Figure 6.12a depicts the overall energy consumption for the base and the smart threshold cases. As can be seen, and, differently to Simulation #1, energy consumption surpasses the base case for prediction thresholds above 100. Due to the building being crowded, the BEMS always maintains indoor elements *on*. Moreover, when a room has been emptied, high prediction thresholds detect future actions and, instead of turning devices *off*, it keeps them *on* for delivering the desired conditions for future occupants. That is the main reason why high smart threshold values offers higher energy consumption than the base case.

Mean occupant comfort, shown in Figure 6.12b, follows a trend very similar to Simulation #1; low prediction thresholds do not grant better comfort than the base case. Specifically, threshold 50 technically delivers the same comfort as the 94.92% of the base case. As the threshold increases, smart cases surpass the base comfort, arriving at a maximum of 97.61% comfort.

As for the success delivery rate plotted in Figure 6.12c, it can be seen that smart cases always set more rooms in the desired status, due to the possibility to predict actions. However, it can be seen that, due to the high number of occupants, the difference between the base and smart curves has been reduced with respect to Simulation #1. In particular, base case reaches 88% whilst smart thresholds vary between 93% and 96%, with only a 5-8% gap.

6.4.3 Simulation #3: Uncrowded Class Building

This simulation is characterised by the specification of a building layout mainly composed of classrooms, with few meeting rooms and office rooms. Table 6.8 defines the parameters utilised for the specification of the building layout and the number of occupants of each type.

Parameter	Value
Offices	5
Meeting rooms	2
Classrooms	40
#Professors	10
#Students	100
#External personnel	5

TABLE 6.8: Building structure and number of occupants for Simulation #3.

As can be seen, the difference between this simulation and Simulation #2 lies in the number of occupants. This variation is performed in order to check whether monitored metrics vary their tendency depending on the number of occupants.

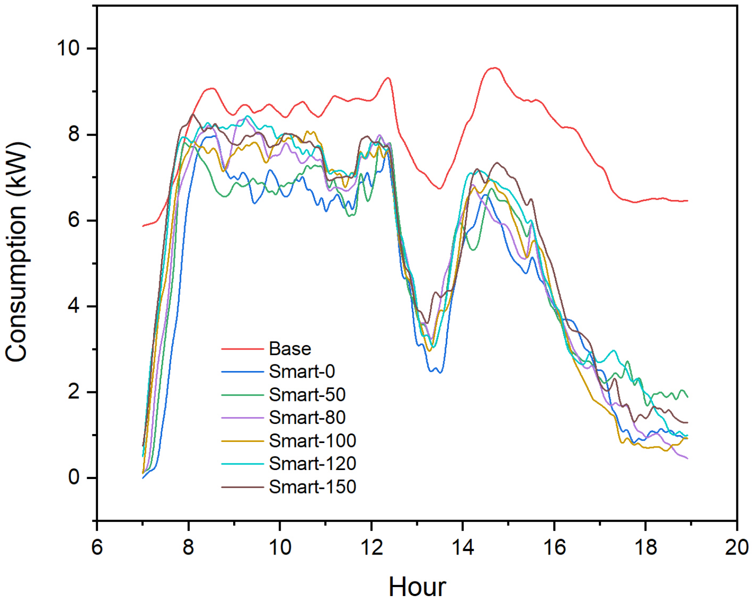


FIGURE 6.13: Daily energy consumption evolution for base and smart threshold scenarios for Simulation #3.

Figure 6.13 plots the daily energy consumption evolution for the base case and the different smart threshold values. Equally to Simulation #1, the gap between the base and the smart cases during the morning is maintained low. Even so, smart cases guarantee less energy consumption. Even with a building with many shared classrooms, it can be seen that the BEMS is able to optimise the resources during lunch hours and the afternoon. During exit hours, the BEMS is able to increment the gap with the base case by actuating upon empty room elements.

In summary, this scenario delivers very similar results to Simulation #1, which allows us to conclude that the building layout has less impact than the actual number of occupants and their movements.

Threshold	**Energy Consumption (kW)**	**Mean Comfort (%)**	**Success Rate (%)**
Base	1,149.5	93.72	76.62
Smart 0	668.06	90.04	92.25
Smart 50	723.41	92.53	92.91
Smart 80	739.69	94.13	93.41
Smart 100	740.04	94.9	93.9
Smart 120	798.98	96.19	93.98
Smart 150	803.19	96.79	94.34

TABLE 6.9: Energy consumption, comfort and success comparison between smart and base scenarios for Simulation #3.

As in the previous simulations, Table 6.9 summarises the overall metric values obtained for the different cases for energy consumption, mean occupants' comfort and room success delivery rate. Figure 6.4.3 plots such values in order to extract conclusions clearly.

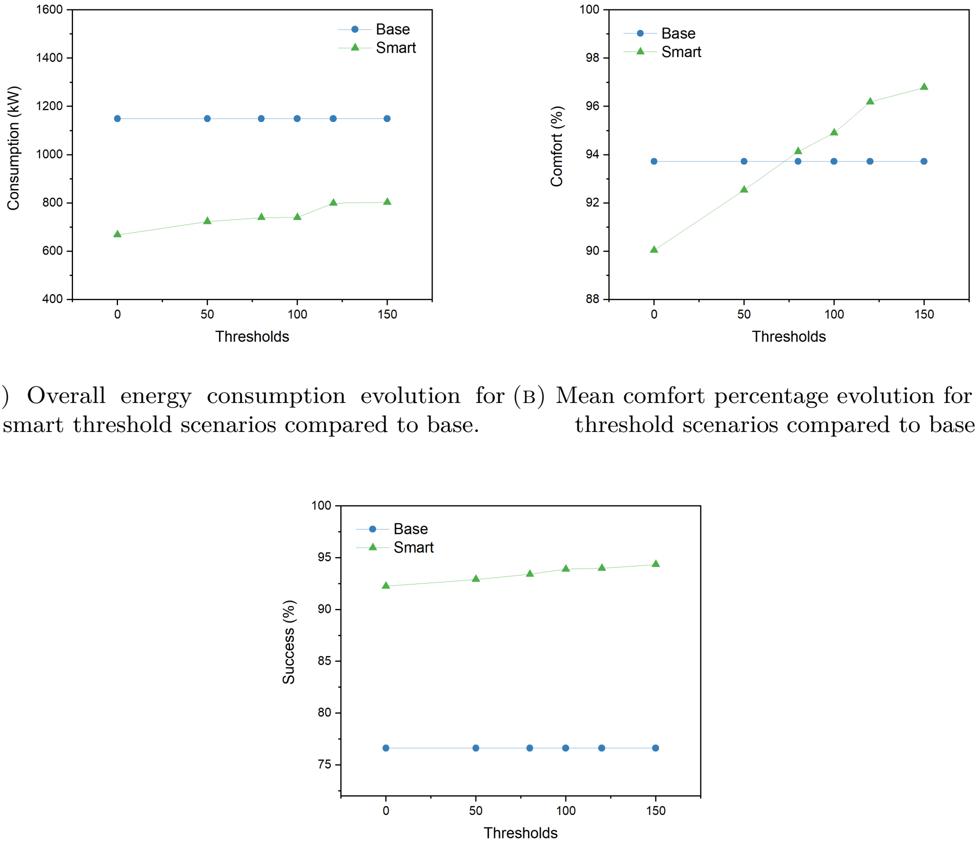
Figure 6.14a depicts the overall energy consumption evolution for the base and different smart threshold values. Similarly to Simulation #1, smart energy consumption always lies below the base case.

In the case of comfort, shown in Figure 6.14b, it presents a very similar tendency to that of Simulation #1. For smart thresholds below 50, comfort is not able to reach the 93.72% of the base case, and they can only reach 92.53% comfort. As the threshold increases, base case is surpassed and the smart case can deliver a comfort of up to 96.79%.

Regarding success rate, presented in Figure 6.14c, smart thresholds make the success rate increase to up to 94.34%, very ahead of the 76.62% achieved by the base case. Moreover, it is worth noticing that this simulation delivers the maximum gap (15-18%) between base and smart cases. This is mainly due to the high number of shared rooms present in the building. Since shared rooms have a high flood of people, smart predictions allow for maintaining indoor elements always in the desired state, whilst the base case actuates upon them even if another occupant is about to enter.

6.5 Conclusions

Results for the different simulations have shown that the enhancement of a building with smart management capabilities can actually deliver better energy optimisation and occupant



(A) Overall energy consumption evolution for smart threshold scenarios compared to base.

(B) Mean comfort percentage evolution for smart threshold scenarios compared to base.

(C) Room delivery success evolution for smart threshold scenarios compared to base.

FIGURE 6.14: Energy consumption, comfort and success comparison between smart and base scenarios for Simulation #3.

comfort increase. Moreover, it has been seen that the number of occupants impacts more than the building layout in the energy consumption outcome. In particular, Simulations #1 and #3 have depicted that, with low number of occupants, the BEMS is capable of reducing energy consumption when compared to the base case for the different smart threshold values. However, a minimum level of predictability is needed in order to guarantee that occupants increment their comfort with respect to the base case. A prediction threshold of 50, i.e. 8 minutes, fails at deliver better comfort, so increasing such threshold to 100 allows to grant occupants better indoor scenarios.

Simulation #2, in which occupant flood is higher and more constant, shows difficulty when trying to optimise energy consumption. However, it is successful at delivering similar consumption values. Nonetheless, with similar consumption values, the BEMS is able to deliver better occupants' comfort for smart thresholds above 50.

It can be concluded that, under our hypothesis and assumptions, and, independently of the building layout and the number of occupants, the BEMS is able to improve current scenario by guaranteeing better energy efficiency whilst increasing occupants' comfort when a prediction window of 10-15 minutes is utilised.

Additionally, the trade-off between energy efficiency and comfort is also present throughout all the simulations. As the comfort of the occupants increases, so does the energy consumption of the building. Due to this, the utilisation of a software to manage building resources permits the dynamic switching between building optimisation modes to deliver a comfort-aware or an energy-aware scenario, in the event of only needing to optimise a single metric. That is, high energy efficiency or high occupants' comfort can be set depending on the prediction parameters.

Chapter 7

Conclusions and Future Work

7.1 Concluding Remarks

Since the beginning, this thesis has tried to deeply study the new Internet of Things paradigm responsible for helping in the optimisation of resources in the Future Internet. We have presented a novel Cloud-based IoT architecture, at its time, that solves the vertical silos problem highly present in the early development stages of this new Internet field. Moreover, the utilisation of Big Data techniques in the Cloud guarantees close to real-time information delivery, crucial for the correct functioning of Management Systems.

Thanks to IoT, Wireless Sensor Networks have gained momentum and their utilisation in this new field is mandatory. However, previous WSN optimisation criteria falls short when it comes to IoT. Due to the necessity to guarantee that no data is lost, it is important to provide the network with backup capabilities in order to ensure that, in the event of partial network failures due to unexpected problems, sensors are able to re-route their data throughout safer paths.

For this, the thesis has studied protection techniques for both homogeneous and heterogeneous WSNs. In the case of homogeneous networks, the thesis has presented a GRASP meta-heuristic capable of detecting the most important combination of sensors inside large networks. Such importance is detected by calculating the degradation in latency and lifetime when eliminating them. Results have shown that, even though meta-heuristics often fail at delivering optimal results, our contribution offers good solutions when compared to an exact Integer Linear Program. Moreover, the introduction of a custom set of features for pruning the tree of possible input sets of critical sensors permits obtaining results for large networks, infeasible for ILPs.

Regarding heterogeneous networks, the thesis has firstly focused in the placement problem widely studied during many years. The problem solves the optimality sensor placement inside a given area. However, IoT requires additional constraints due to the necessity to specify which type of sensors must be collocated at every section of the physical layout, in order to sense and acquire surrounding metrics correctly. For this, we have extended the placement problem by introducing clustering constraints. The problem has been formulated using an ILP and it has been tested in an office building layout. Additionally, protection is granted by means of a protection level constraint that mandates the level of available transmission paths each sensor must have, in order to re-route their data in the event of unexpected partial network failures. Results show that, even with additional constraints, the problem still delivers better results than a complete sensor deployment. The protection level, thought, presents a clear trade-off with respect to energy consumption. That is, the higher the protection, the higher the consumption.

In both cases, by optimally placing sensors and by delivering backup capabilities to the crucial part of the network, it is possible to manage resources efficiently and avoid wasting energy unnecessarily.

Finally, we aimed at utilising the WSN placement results inside an office building to develop a real case inside the university, in order to feed a Building Energy Management System that optimizes building resources. However, due to the high cost and security flaws it might have, it was decided to create a simulation tool for mimicking such behaviour.

The simulation tool aims at efficiently manage energy and increase occupants' comfort. However, conclusions have shown that there exists a clear trade-off between the two metrics. Nonetheless, the management of building resources thanks to a software defined layer, permits the dynamic switching between metrics. Results for three different building layouts and number of occupants have shown that the number of occupants is more important than the actual building layout, always inside our hypothesis and models presented.

When the number of occupants is very high, the BEMS cannot manoeuvre enough to optimise resources. Instead, similar energy consumption as the current building behaviour is obtained. Nevertheless, thanks to its predictability, it is possible to increase occupants' comfort. For buildings with less occupants and more room to manoeuvre, the BEMS shines by drastically reducing energy consumption since it avoids wasteful scenarios. Moreover, results have shown that, even with a prediction window of 25 minutes, it is still possible to increase occupants' comfort whilst maintaining acceptable levels of energy consumption.

7.2 Future Work

Further work in the research line of this thesis can be performed by extending our current models. Due to the high growing speed of IoT, constant evolution is needed in order to adapt to new solutions and requirements. Particularly, architectures are constantly evolving and new studies must be performed in order to check whether future systems might require an adaptation of our presented work.

The simulation tool permits several improvements. Particularly, occupant behavioural models can be extended to other areas in order to permit the simulation of different types of buildings. Moreover, new environmental variables can be introduced and new types of actuation upon different indoor elements can be specified. The cognition system, responsible for launching actuations upon indoor elements, can be extended by introducing Machine Learning techniques in order to learn from the past, and offer higher resource optimisation and occupants' comfort as more knowledge is acquired. Inside BEMS, this is still a challenge due to the difficulty to integrate many heterogeneous metrics inside a Machine Learning algorithm, and draw clear conclusions about what actuation path to follow.

Furthermore, BEMS results and conclusions drawn from our simulations must be confirmed against a big enough real testbed. However, the utilisation of BEMS is not restricted to the inability to deploy a real system, since simulation tools grants high flexibility and testability.

Appendix A

Published Work

A.1 Publications in Journals

Ricciardi, S., **Sembroiz-Ausejo, D.**, Palmieri, F., Santos-Boada, G., Perelló, J., and Careglio, D. "A hybrid load-balancing and energy-aware RWA algorithm for telecommunication networks". In: *Computer Communications* 77 (2016), pp. 85–99. ISSN: 0140-3664. Impact factor: 3.338, Q1. DOI: 10.1016/j.comcom.2015.06.010.

Sembroiz, D., Careglio, D., Ricciardi, S., and Fiore, U. "Planning and operational energy optimization solutions for smart buildings". In: *Information Sciences* 476, pp. 439–452. ISSN: 0020-0255. Current Impact factor: 5.524. DOI: 10.1016/j.ins.2018.06.003.

Sembroiz, D., Ojaghi, B., Careglio, D., and Ricciardi, S. "A GRASP Meta-Heuristic for Evaluating the Latency and Lifetime Impact of Critical Nodes in Large Wireless Sensor Networks". In: *Applied Sciences* 9.21 (2019). ISSN: 2076-3417. Current Impact factor: 2.217. DOI: 10.3390/app9214564.

A.2 Publications in Books

Sembroiz, D., Ricciardi, S., and Careglio, D. "Chapter 10 - A Novel Cloud-Based IoT Architecture for Smart Building Automation". In: *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks.* Ed. by Massimo Ficco and Francesco Palmieri. Intelligent Data-Centric Systems. Academic Press, 2018, pp. 215 - 233. ISBN: 978-0-12-811373-8. DOI: 10.1016/b978-0-12-811373-8.00010-0.

Appendix B

HyLERA. A Hybrid Load-balancing and Energy-aware RWA Algorithm for Telecommunication Networks

B.1 Introduction

In the last years the Internet traffic has been rising according to an exponential trend, leading to increased bandwidth demands which in turn has resulted in more sophisticated and energy-hungry communication equipment that affect significantly the networks' operational expenses. At the state of the art, telecommunications networks infrastructures require more than 1% of the worldwide production of electrical energy and the demand rate increases of about 12% per year [89]. These numbers give an immediate idea of the impact of the energy bills on the overall economy of most of the large-scale network communication provider organisations. The growth of energy demand together with the limited availability of new and more clean energy sources also introduces significant issues in terms of ecological impact. Until most of the energy needed for network operations will not be drained from renewable and "green" energy sources, a large amount of Greenhouse gases (GHG) will continue to be emitted into the atmosphere as a consequence of the direct and indirect usage of the communication equipment. This implies a growing attention to the ecological footprint [90] of network infrastructures that is an essential prerequisite for sustainable development [91] of the modern network-centric society. Furthermore, the heavy usage of "dirty" energy sources, apart from the expected effects on the energy bill, may also be origin of additional costs in presence of specific "carbon containment frameworks", such as *carbon taxes*, *cap and trade* or *carbon offset* [92], whose objective is encouraging the use of green sources. This implies that an additional costs, ideally compensating the adverse effects on the environment, is introduced if the drained energy, derived from sources characterised by high GHG emissions, exceeds a specify threshold.

The only viable strategies for containing the power consumption and consequently the carbon footprint in modern communication networks rely on the energy proportional behaviour of most of the new generation equipment, that are able to adapt their own energy demand to the effective workload by dynamically switching between several operating states, each characterised by a higher or lower component performance (e.g., interfaces, memories, switching fabric, etc.). These strategies require the introduction of energy-awareness in the network control plane, providing visibility of the energy efficiency degree of all the network equipment in order to route communications on paths traversing switching devices and links characterised by a lower proportional absorption, and hence minimizing the load on the most energy-hungry devices.

However, this may introduce several undesirable side effects on the overall network optimisation policies and objectives. That is, while being effective in substantially reducing the energy costs and/or associated ecological footprint, routing strategies that strive to divert connections over the paths requiring minimum energy, without considering all the other more traditional traffic engineering objectives, may result in almost blind choices that tend to unbalance the network load and under-utilise many expensive communication links, with obvious consequences in terms of return of investment (ROI).

From the previous considerations, it is immediately evident how control-plane strategies aiming at balancing resource utilisation or containing the energy consumption as well as GHG emission, can easily become mutually contradictory. To cope with this problem, we propose a Hybrid Load-balancing and Energy-aware routing and wavelength assignment (RWA) algorithm (*Hylera*) whose goal is to combine, according to a threshold-based scheme, the aforementioned energy demand optimisation and resource usage balancing strategies in order to achieve more stable effects on the overall network engineering economy, by using *at any time* the most appropriate strategy. Thus it drives lightpath selection based exclusively on a load balancing objective when the risk of request blocking grows over a certain threshold (i.e., the network is experiencing an overload condition), and it performs its connection routing and wavelength assignment choices in order to reduce the overall energy consumption when the network is unloaded.

The rest of the work is organised as follows. In the related work section we analyse the RWA experiences already available in literature that follow load-balancing or energy-aware approaches as well as those ones which try to unify or hybridise both of them. The next section defines the methodology proposed in Hylera and the following one analyses its performances through simulation and discusses the results obtained. Finally, the last section reports the conclusions and the future research plans.

B.2 Energy-aware RWA in WDM networks

Network Providers typically design and manage their own transport infrastructures in order to attract an ever increasing interest in their customers by providing high bandwidth, extremely reliable and quality-differentiated communication services. At the same time, they incur in significant initial investments and operating costs so that they are continuously faced with the challenge of using the most expensive equipment and communication resources as efficiently as possible, by keeping their power consumption costs at a minimum and using all the available optical links in a fairly balanced way. Unfortunately the last two objective are often in contrast, depending on the specific network topology and equipment used. However, energy expenses, due to their recurring nature become one of the most critical element in the overall operational costs. Consequently, deploying dynamic power management strategies that aim at decreasing the power demand in the operational phase as well as at reducing energy bills by exploiting the use of renewable energy sources or taking advantage from time/location-dependent fluctuations in energy costs, becomes a fundamental prerequisite for maximizing their medium and long-term revenues. Stimulated by the above needs and by recent advancements in traffic engineering techniques, there is a growing interest in designing a unified control plane framework resulting in a smart integrated approach that combines the above resource usage and energy cost containment strategies into an hybrid optimisation framework whose main aim is achieving the best compromise between the apparently disjoint (or, worse, conflicting) load balancing and and energy-related objectives, by harmonizing them into a single revenue-maximisation goal.

B.2.1 Circuit switching in multi-layer optical networks

A multi-layer optical network is a very complex mesh of variously interconnected heterogeneous sub-networks composed of an electronic IP layer (network edge), providing network access and connectivity distribution, built on top of an optical transport layer (OTN) playing the role of interconnection backbone (network core). Each sub-network consists of multiple heterogeneous switching devices, ideally operating according to a common control plane protocol and management policy, usually within the same Autonomous System (AS). In presence of very different types of devices, built by multiple vendors, all the switching decisions are based on a combination of packet, time slot, wavelength or interface depending on the location (edge or core) and role (intermediate or terminating) of the involved devices within the overall network topology.

Generalised multi-protocol label switching (GMPLS) is the control plane solution of choice that is used in most of the cases to automatically route the connections from source to destination, possibly crossing multiple fibre spans. When the connections are subject to some quality of service (QoS) requirement or impairment constraint – e.g. required bandwidth, latency, BER, etc. – all the individual links involved in the generalised labelled switched path (LSP) – becoming a lightpath in the optical layer – have to satisfy the above requirements/constraints so that the whole path has to be determined according to a traffic or network engineering approach (this is usually referred to as MPLS-TE).

At the optical layer, wavelength division multiplexing (WDM) technology is used to multiplex several channels – each on an individual wavelength – on the same fibre. If no wavelength converters are present in the network, a connection has to be allocated on the same wavelength in the optical domain (wavelength continuity constraint, WCC), and obviously two connections sharing the same fibre have to use different wavelengths (clash constraint). Since a single wavelength can normally carry a channel of 40 or even 100 Gbps, it is common to multiplex several tributary sub-channels in one wavelength (a process known as grooming); the wavelength resource is thus divided in time slots which will be assigned to the individual sub-channels, possibly with different occurrence distribution (in the case of statistical multiplexing).

The WCC can be relaxed by providing the optical cross connects (OXC) with wavelength converters, which are quite expensive devices; a more economically viable alternative is to extract the wavelength from the fibre, with a reconfigurable optical add and drop multiplexer (ROADM), and convert it in the electronic domain from which it can be reintroduced in the optical domain using a different wavelength – a process known as O-E-O conversion. However, due to the limiting processing speed of the electronics, such a process is usually avoided and limited only to the case in which a 3R regeneration (re-amplification, re-shaping and re-timing) of the optical signal is required in order to preserve the carried information with an acceptable bit error rate (BER).

When a connection request reaches a lambda edge router (LER), a feasible path composed of multiple communication links and nodes connecting the request source with its destination, together with an appropriate wavelength to be used on it, have to be selected in order to setup a dedicated end-to-end communication channel (circuit). Normally, there are several paths that can be selected with enough free bandwidth and satisfying the connection's QoS requirements. The final choice between the available options depends on the routing and wavelength assignment (RWA) algorithm and its optimisation objectives.

B.2.2 RWA with hybrid optimisation behaviour: the HyLERA idea

Let us consider the network depicted in Figure B.1. Suppose that a request has been issued in order to connect two nodes at the IP layer. The source node, which has a complete view of the network provided by the internal routing protocol of the involved AS (such as OSPF-TE), looks for a feasible path at the optical transport layer. It is easy to see that two feasible paths are available, namely lightpath A and lightpath B (suppose there is a free wavelength or sufficient free bandwidth on both in case of grooming). Lightpath A is shorter (two hops at the optical layer), and traverses an unloaded node (i.e., currently switching few or no connections); lightpath B is longer (three hops at the optical layer), but traverses two energy-efficient nodes. A pure load-balancing algorithm would select lightpath A, whereas a pure energy-aware algorithm would select lightpath B. The idea of the HyLERA algorithm is to select one or the other path according to the *current* network state/operating conditions. If the network is unloaded (e.g., at night or during low load periods), it prefers optimizing energy-efficiency and hence selects paths requiring less power to be operated, even if longer in terms of traversed hops/nodes or quite unfair in terms of traffic load distribution (note that selecting a longer path will consume more resources for serving the same connection request compared to a shorter path). When the network becomes more loaded (i.e., more connections requests come into the network, for example during daytime or peak traffic

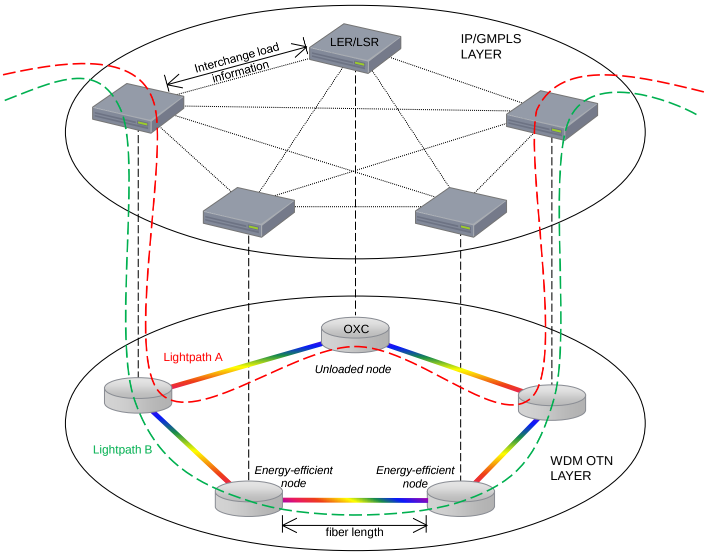


FIGURE B.1: Path selection in a multilayer GMPLS/WDM network according to a pure load-balancing (red line) or a pure energy-efficient (green line) criteria; the optimal selection depends on the current state of the network.

hours), the algorithm automatically switches to load-balancing mode, preferring the more fairly-balanced lightpaths in order to save resources and serve the highest possible number of connection requests, eventually coming back again to energy-efficiency mode later, during lower load periods. The switch-over among the two criteria is driven by a threshold on the arrival rate of connections requests measured during a (parametric) sliding window. The rationale to use connections requests arrival rate and not directly the blocking rate is that it is difficult to find an appropriate value of the blocking rate that is effective regardless the network topology, the time of the day, the traffic pattern and the state of the network. The connections arrival rate is instead an easily tunable parameter for a network, regardless of the "accumulated traffic", i.e. of the current load of the network, and of the future incoming traffic. In this way, starting from the knowledge of the recent past (provided by the sliding window), we estimate the near future and accordingly operate the network in the most efficient way.

It is worth to underline that an edge cost function that mixes energy and load in one function is not a good option due to the different optimisation objectives that are in contrast; therefore, we prefer an hybrid solution that however is forced to act *either* load-balancing *or* energy-efficiently, according to the what is currently most convenient, given the current configuration of the network. This also provides simplicity to the HyLERA algorithm, though its effectiveness and ease of use.

B.3 The energy model

In order to handle the energy-awareness objectives, we need the ability to estimate for each candidate connection, in addition to the more traditional link or node properties that lead to the determination of the shortest and/or less congested paths, also the specific metrics

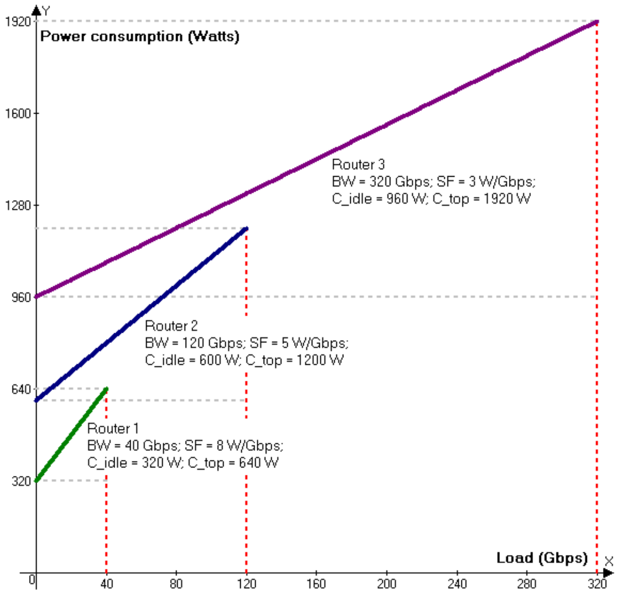


Figure B.2: The energy consumption of three routers with different aggregated bandwidth (BW) and scaling factors (SF) are represented.

that describe their energy consumption. A linear energy model has been used to describe the energy consumption of routers at different loads. According to real router measurements [93], we consider that, when a router is turned on but *idle*, it consumes half of its total power consumption and, as the load increases, its power consumption linearly increases, up to its maximum value which is reached when the router is fully loaded. The slope with which the power consumption function increases with respect to the load is given by its energy *scaling factor*, measured in W/Gbps in the the Energy Scaling Index (ESI)[1]. A scaling factor of x W/Gbps means that x W are required to route 1 Gbps of traffic. Typical values for the scaling factor vary from 1 to 10 W/Gbps [89], where small routers consume more energy per bit than larger routers, since the latter ones are typically more efficient and tend to be placed in the core of the network where the traffic is more aggregated [94].

In Figure B.2, the energy consumption of three different routers [89] has been drawn. Router 1 is a small router with a maximum capacity (aggregated bandwidth of all its interfaces) of 40 Gbps, and has a scaling factor of 8 W/Gbps; therefore, it consumes a maximum of 640 Watts (320 W just to stay ON, and 8 × 40 W when fully loaded). Similarly, Router 2 is a medium router (120 Gbps of aggregated bandwidth) with a scaling factor of 5 W/Gbps, and Router 3 is a big router (320 Gbps) with a scaling factor of 3 W/Gbps. It is worthwhile to note that the power consumption of idle routers is always present since we assume that no sleep mode is available at router level, i.e. an idle router can not be put into sleep mode, as reported also in [95][96]. Therefore, any energy-related optimisation relies on the variable power consumption of routers, selecting short routes passing through routers with low energy scaling factors.

[1] The Energy Consumption Rate (ECR) and the Energy Scaling Index (ESI) are equivalent metrics used to measure the efficiency of routers, where the former uses energy per bit (nJ/bit) and the latter uses Watts per Gbps (W/Gbps); in fact, it holds that W/Gbps = (J/s)/(Gbit/s) = J/Gbit = nJ/bit.

Router	**Aggregated bandwidth** $(Gbps)$	**Scaling factor** $(W/Gbps)$	**Power consumption** (W)	**Power consumption** (y) **as a function of Load** (x) $y = f(x)$
Router 1	40	8	$C_{top} = 640W$ $C_{idle} = 320$	$y = 8x + 320,$ $x \in [0, 40]$ $y \in [320, 640]$
Router 2	120	5	$C_{top} = 1,200W$ $C_{idle} = 600$	$y = 5x + 600,$ $x \in [0, 120]$ $y \in [600, 1200]$
Router 3	320	3	$C_{top} = 1,920W$ $C_{idle} = 960$	$y = 3x + 960,$ $x \in [0, 320]$ $y \in [960, 1920]$

Table B.1: The values used for the three routers: Aggregated bandwidth (BW), Scaling factors (SF), Power consumption when fully loaded (C_{top}) and in idle (C_{idle}), and finally the equations describing the Power consumption (y) as a function of the Load (x).

Table B.1 summarises the parameters of the three routers and provides the equations describing their power consumption as a function of the load.

In general, the power consumption (y) of a router as a function of its load (x) is defined and takes value over the following domain and image:

$$y : [0, BW] \to [C_{idle}, C_{top}], \tag{B.1}$$

and it is given by the following equations:

$$y = SF \cdot x + C_{idle}, \tag{B.2}$$

where BW is the router aggregated bandwidth, C_{idle} and C_{top} are respectively the power consumption of the router when idle and when fully loaded and SF is its energy scaling factor. 3R regenerators have an energy scaling factor SF_{3R} of about 3 W/Gbps, as reported in [89].

B.4 The HyLERA cost function

The HyLERA algorithm is based on the Dijkstra shortest path algorithm, modified to operate in WDM networks and constrained on the availability of enough free bandwidth/wavelengths. The wavelength continuity constraint is imposed along a lightpath when no wavelength converters are present or no O-E-O conversion is performed.

The network is represented as a multigraph $G = (V, E)$, where V, $|V| = n$ is the set of nodes (either electronic router or optical switches/cross-connects), E, $|E| = m$ is the set of edges modelling the links in the network. Note that, since WDM is deployed in the optical network, there can be more than one edge between a pair of nodes (therefore, G is a multigraph), each one representing a wavelength channel.

Each edge $(u, v) \in E$ in the network graph has nonnegative cost $c_{(u,v)} \in \Re^+$ determined by the edge weighting function currently used by HyLERA (either load-balancing or energy-awareness, depending on the current network status). The maximum capacity of each link $b_m(u, v) \in B$, where B is the set of possible bit-rates (310 Mbps, 622 Mbps, 2.5 Gbps, 10 Gbps, 40 Gbps, 100 Gbps), is given by the bit-rate of the interfaces operating on each wavelength; the current available capacity $b_r(u, v)$ of a link is updated each time a connection is set up or torn down in the network.

Thus, network links may be weighted according to pure load-balancing or pure energy-awareness function, depending on the current state of the network. In detail, when the network is unloaded, for example during the night, the links are weighted with the energy-aware function c_{EA}, which makes the Dijkstra algorithm select the lowest energy-consuming links:

$$c_{EA}(u, v) = SF(u) + SF(v) + SF_{3R}(u, v) \tag{B.3}$$

where $SF(i)$ is the energy scaling factor of node $i \in V$, and $SF_{3R}(u, v)$ is the scaling factor of 3R regeneration on edge (u, v). Note that no sleep mode is allowed for the optical amplifiers, therefore they are not evaluated in the equation.

Alternatively, when the network is loaded, for example during peak day hours, the links are weighted according to the load-balancing function c_{LB}, which makes the Dijkstra algorithm prefer the less congested links in an effort to accommodate as much connections as possible.

$$c_{LB}(u, v) = [b_r(u, v) \cdot \log(b_m(u, v))]^{-1}. \tag{B.4}$$

The more the available bandwidth and channel bandwidth are, the lower the cost is. Note that the maximum capacity is weighted with the logarithmic function, in order to lessen its importance with respect to the residual capacity.

The *mode* in which the algorithm is working is identified by a parameter w which can be either *energy-awareness* or *load-balancing.* The switching between the former or the latter function is discriminated by a threshold evaluated in a sliding window. Specifically, the number ξ of connections requests that arrive in the network are monitored during the last k hours (sliding window) and, if they overcome the fixed threshold, the links cost function is changed. In order to better bias the edge cost function to be used at each time, two thresholds are defined (one for load-balancing and the other for energy-awareness): t_{high} and t_{low}. Starting from a void network (i.e., in the initial state), the algorithm works in energy-saving mode ($w =$*energy-awareness*), using the energy-aware function to weight the network links. As connections arrive at the network, they are continuously monitored and, if their number ξ exceeds the t_{high} value during the sliding window, the link cost function is switched to load-balancing ($w =$*load-balancing*) and stays until the connections arrival rate decreases to a value lower than the t_{low} threshold. When this occurs, the link cost function is switched back to energy-awareness to save energy during the low load period. Therefore, the link cost function is defined as:

$$c_{(u,v)} = \delta \cdot c_{EA} + (1 - \delta) \cdot c_{LB}, \tag{B.5}$$

where

$$\delta = \begin{cases} 1 & if\ \xi \leq t_{low} \text{ and } w = load{-}balancing \\ 0 & if\ \xi \geq t_{high} \text{ and } w = energy{-}awareness \end{cases} . \tag{B.6}$$

B.4.1 OSPF-TE extension

The hybrid behaviour of Hylera reaches its maximum effectiveness if all the routers in a network switch at the same time, i.e. in a coordinated, network-wide manner. Therefore, it is necessary to have a distributed mechanism to keep track of all the connections that arrive in the network during the sliding window, representing the instantaneous network load information needed to adaptively drive the switch-over between load balancing and energy-aware behaviour.

To this end, we extend the OSPF-TE Opaque Link State Advertisement (LSA) messages to include the arrival rate information during the sliding window, and spread this information with regular OSPF updates to all routers in the network.

In particular, we add a new Type-Length-Value (TLV) field to the Traffic Engineering extensions for OSPF-TE (Traffic Engineering LSA, opaque type=1) according to an agile implementation (the detailed explanation of each field of the TE LSA can be found in [97]). The *Type* field (16 bits) contains the ID of the newly defined entry (32,768, which is the first available one) and the *Length* field (16-bits) specifies the extension of the the *Value* field (in octets), which are contained in the payload of LSAs. As reported in Table B.2 the Value field identifies the current connection arrival rate experienced by the involved device.

Type	**Length**	**Value**
32,768	4 octets	Connections arrival rate

Table B.2: Sub-TLV for TE LSA.

Note that the the scaling factors (SF) used in the OSPF metric, are statically defined at the network definition time, and it is not necessary to spread them periodically. However, it is also possible to configure OSPF to automatically spread them in case, for example, of changes in the topology; to this end, it is sufficient to add two other 4-octets sub-TLV fields, namely 32,769 and 32,770, containing respectively the node and the 3R SFs.

It is also worth to note that, even though OSPF is a link-state protocol (i.e., a flood only if a change), a link state refresh time (*LSRefeshTime*) is defined; when this time expires, a router floods a new LSA to all its neighbours, who will reset the age of the sending router's records to the new received age. OSPF sets the *LSRefreshTime* to 30 minutes [98], which is lower than the sliding window timespan (ranging from 1 to 6 hours). Therefore, the periodic update done by the OSPF is directly usable to spread the number of connection requests arrived during the sliding window and the proposed TE LSAs will be flooded over the whole network on such a fixed time-basis, disseminating the network load information.

B.5 Time and Space Complexity Analysis

The HyLERA algorithm is illustrated in Figure 8. It takes as input the graph representing the network, the connection request with source and destination nodes and the required bandwidth, the high and low thresholds and the connection requests count in the sliding window. The edge cost function (Load Balancing / Energy Awareness) initially is set to energy-awareness (since the network is void).

Algorithm 8 Hylera($G, c, t_{high}, t_{low}, \xi$)

```
Require:
  G: current network state
  c = (s,d,b): connection request; s, d: source, destination nodes; b: required bandwidth
  $t_{high}$ : ascending threshold to change from Energy Awareness to Load Balancing
  $t_{low}$ : descending threshold to change from Load Balancing to Energy Awareness
  $\xi$: number of connection requests in the sliding window
Ensure:
  $G^*$: new network state
  $\pi^*$: new lightpath
1: if $w$ =energy-awareness AND $\xi \geq t_{high}$ then
2:     $w \leftarrow$ load-balancing
3: else if $w$ =load-balancing AND $\xi \leq t_{low}$ then
4:     $w \leftarrow$ energy-awareness
5: end if
6: $\pi^* \leftarrow$ constrainedDijkstra(G, c, w)
7: $G^* \leftarrow$Update the $w((u,v)_\lambda)$ costs of network edges $(u,v)_\lambda$ along the chosen path $\pi^*$ and
   increment $\xi$ of one unit
8: return $(\pi^*, G^*)$
```

Lines 1-5 discriminate what is the edge cost function w that has to be used, given the current network state, i.e., the number ξ of connections in the sliding window. This is performed by a simple check between ξ and the values of the thresholds and possibly the consequent assignment. These lines have a constant time complexity of $O(1)$.

The HyLERA algorithm is based on the Dijkstra's algorithm, modified to operate in WDM networks and constrained on the availability of enough free resources to serve the connections. The constrained-based routing is performed in line 6, checking, when a new node is discovered by the algorithm, if the available bandwidth on the link connecting that node is equal or greater than the connection request required bandwidth. The WDM network is represented as a multigraph, in which there can be more than one edge between a pair of nodes, representing the different WDM channels. Therefore, when a new node at the lowest distance from the source is discovered, if there is enough free bandwidth, it is labelled not only with distance from the source and predecessor node, but also with the predecessor wavelength on which the node was reached. Both the bandwidth constraint check and the additional labelling have a constant time complexity of $O(1)$. The Dijkstra algorithm has a computational complexity of $O(m + n \log n)$, when improved by using a priority queue with a Fibonacci heap in the implementation [99].

In line 7, the new network state (i.e., the edges residual bandwidth and costs) is updated only for the network edges involved by the new path π^*, which, in a network with n nodes, has a maximum length of $n - 1$, therefore having a computational complexity of $O(n)$.

Finally, the new path π^* and the new network state G^* are returned in line 8.

Therefore, the HyLERA computational complexity in the worst case scenario is $O(m + n \log n)$ which is the same as the original Dijkstra algorithm.

As for space complexity, the multigraph network representation employed by HyLERA requires less space with respect to the layered graph approach conventionally used in dynamic RWA algorithms [100]. Using up to λ wavelengths on each edge, the layered representation with C converter nodes will require $\lambda n + 2$ nodes (λ layers, each dedicated to an individual wavelength, plus two additional nodes to serve as ingress and egress) and $\lambda m + 2\lambda + C \cdot (\lambda - 1)$ edges (converters can be modelled by cross-layer edges that connect each layer to the λ adjacent layer – a wavelength conversion spanning multiple frequencies will thus entail many such edges in sequence), whilst the equivalent multigraph representation will require only n nodes and λm edges, thus notably reducing the space complexity. Besides, in the layered graph, the ingress and egress nodes as well as the edges connecting them to the network have to be built each time a new connection arrives, whilst in the multigraph approach this preprocessing phase is not necessary thanks to its compact representation. Note that, even in absence of wavelength conversion, all the layers of the layered graph have to be explored, since the (first) wavelength of the lightpath may be any, which compensates the additional check needed in the multigraph approach to enforce the wavelength continuity constraint. Furthermore, the higher number of nodes and edges required by the layered graph with respect to the multigraph approach increases the time complexity which strictly depends on the n and m parameters.

B.6 Performance Evaluation

Extensive simulation experiments have been performed in order to evaluate the efficiency of HyLERA under different operating conditions and scenarios. In order to resemble closest-to-reality scenarios, we made specific assumption on the traffic pattern, the distribution of connection requests and on the network topology and node design.

For this purpose, we modelled the well-known Geant2 pan-European research network. The network has 34 optical switches, each connected to an electrical router. The links between node pairs are modelled according to the real topology, with a WDM degree and interfaces

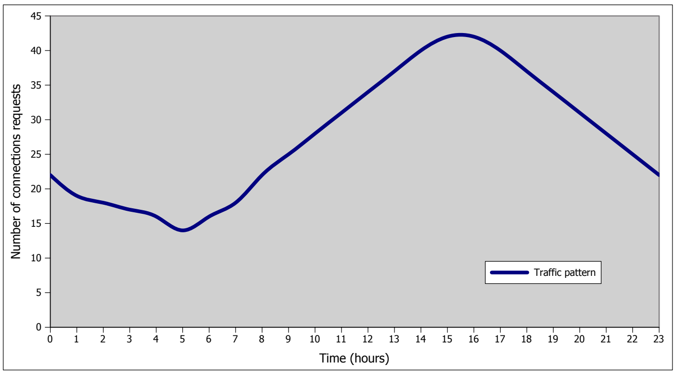


FIGURE B.3: The pseudo-sinusoidal traffic pattern.

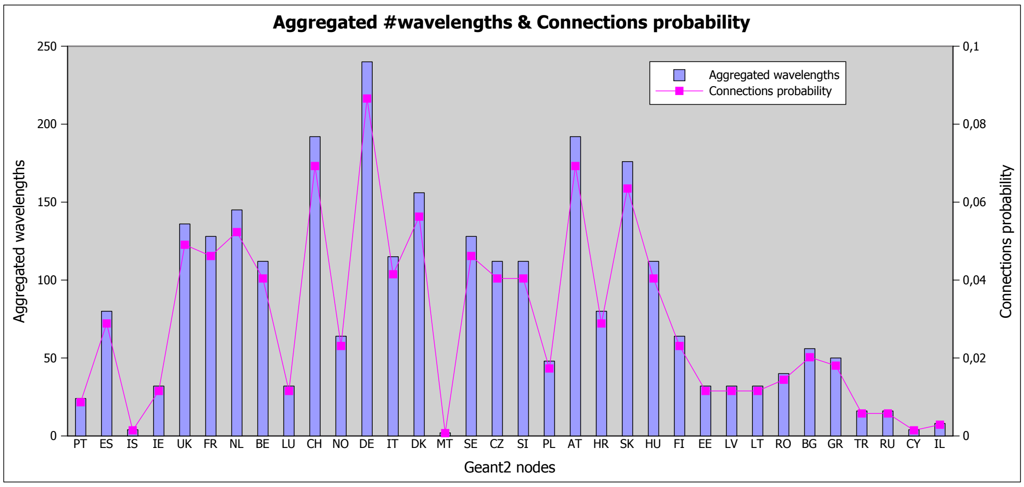


FIGURE B.4: The number of wavelengths of each network node is reported (columns) together with its probability to appear as either source or destination in connection request (dots).

bit-rates scaled up as shown in the figure. The physical distance among nodes is reported in the figure and optical amplifiers (OA) and regenerators (3R) have been put accordingly on each link: an OA each 80 km and a 3R each 500 km. Simulations have been performed on an Intel® Core^{TM} i7-950 CPU @ 3.07 GHz with 16 GB RAM and 64 bit operating system server equipped with the Sun® Java® Runtime Environment v.1.6. To perform the simulations, we used SimulNet [101], an ad-hoc optical network simulation environment that allows flexible modelling of network topologies as well as traffic load generation, data recording and post-processing. Simulation parameters are reported in Table B.3.

We modelled the traffic pattern as pseudo-sinusoidal over the 24 hours of a day, as shown in Figure B.3 and reported in [102].

In order to simulate real traffic matrices, the traffic pattern has not been uniformly distributed over the network nodes: bigger routers have been assigned higher probability to be selected as source or destination of connections requests than smaller ones. Such a probability is therefore linearly dependent on the aggregated number of wavelengths that a router manages, with the distribution probability shown in Figure B.4.

Simulation Parameters	Geant2 Network
Simulated time	4 days
Time steps	2,400 per day (one step every 36 seconds)
Number of generated connections	Varying from 0 to 263,200
Connections lifetime	30 min
Connections Bandwidth	310 Mbps
HyLERA sliding window k	1, 3, 6 hours
HyLERA threshold t_{high}	Simulation #1: 9,300; Simulation #2: 11,900; Simulation #3: 5,400
HyLERA threshold t_{low}	Simulation #1: 8,400; Simulation #2: 10,200; Simulation #3: 5,200
Parameters Λ_{OA}, Λ_{3R}	80 km, 500 km
Distribution of nodes as Source or Destination of connections	Weighted according to the number of wavelengths of the routers (reported in Figure B.4)
RWA algorithms	HyLERA, pure load-balancing SPF, pure energy-aware SPF
Measurements	Blocked connections and power consumptions

Table B.3: Parameters used in the simulations.

B.6.1 Simulations Results

Due to space limitations, we only report a set of three *notable* simulations, that show the flexibility of HyLERA in achieving different optimisation objectives. In the first set of simulation, HyLERA parameters (the high and low thresholds) have been set up in order to obtain the optimal behaviour, given the current conditions, i.e. for the given network topology, traffic pattern and distribution, etc. Since the behaviour of HyLERA is determined by the thresholds, we can force HyLERA to work by using one or the other cost function, either trying to save more energy or trying to block less connections. Therefore, the second and third simulation tests have been performed by setting the parameters in order to make HyLERA behave in a *more* energy-aware or load-balanced way, respectively. In order to evaluate the performance of HyLERA, we compared its results with the ones associated to pure load-balancing and pure energy-aware algorithms, which are shortest path first (SPF) algorithms in which edges are labelled exclusively with the load-balancing (Equation B.3) or energy-aware (Equation B.4) cost functions, respectively. These algorithms provide the lower and upper bounds reference values of the achievable savings and blocking, and therefore the pure SPF algorithms are the same in the three set of simulations and are reported as a comparison. Several time windows have been tested (1, 3 and 6 hours), but only the results for the 3 hours time window are reported, since they give the best trade-off between the past traffic and the reactiveness of the algorithm to changes.

In order to show in a clear manner the results of the simulations, the graphics contain extra Y-coloured-axes. Blue and pink axes represent a change in the edge cost function used by HyLERA, being blue the change from energy-awareness to load balancing, and pink the change from load balancing to energy-awareness. White axes are used to split between simulation days: each simulation has been performed during 4 days (a sufficient time to reach an equilibrium between incoming and outgoing connections in the network, given the traffic pattern of Figure B.3).

Simulation #1: achieving the best-balance between energy-efficiency and load-balancing

The aim of this simulation is to show the flexibility of HyLERA to route as many connections as possible during the highest congestion period (central 12 hours of the day) and to save as much energy as possible during the remaining night hours.

Figure B.5a shows connections blocking probability measured during the sliding window (three hours) during the four days of simulation. During the first day of simulation, the three algorithms (HyLERA, the pure load-balancing SPF and the pure energy-awareness SPF) show an almost equal behaviour, since the network starts empty and the only blocked connections are the ones that have to be established between a source or destination node with very few wavelengths which are already used by previous connections still alive in the network (connections have a mean lifetime of 30 minutes).

In day two, the network is not empty, since some connections are still alive from day one and, when the load increases (at around 8 am), the performance of the three algorithms begins to diverge. The pure energy-aware algorithm will select routes that pass through nodes with lower energy scaling factor (cfr. Equation B.3), which will result in longer, but more efficient, routes. However, selecting longer routes will also consume more resources, since the requested bandwidth is occupied on each link of the route; as a consequence, the connection blocking will significantly increase during central day hours.

On the other hand, the pure load balancing algorithm keeps the connections blocking at much lower rates, since it prefers the routes that pass through nodes with higher available and maximum bandwidths (cfr. Equation B.4). However, these routes are not the most energy-efficient, since the pure load-balancing SPF does not consider the nodes scaling factor in its decision process.

HyLERA, instead, changes dynamically the edge cost function according to the current network load. It starts with energy-efficiency, since at the beginning of day one the network is empty. Then, the first change occurs at the middle of day one, when the traffic increases during the peak hours. Starting from 4 p.m, the traffic decreases, and at 9 p.m. HyLERA switches back to the energy-saving modality. It is worth to note that, even if in day one the network starts empty and therefore there is no appreciable difference in the blocking connections, HyLERA is "working behind the scene", saving energy already in day one (it will be evident when looking at the energy consumption shown in Figure B.5c). As for the blocking, the better behaviour of HyLERA becomes evident starting from day two on. When the traffic load begins to increase (at approx. hour 35), HyLERA switches to the load-balancing cost function, keeping the blocking connections at values close to the pure load-balancing SPF, whilst the pure energy-efficient SPF increases notably the blocking. When the traffic decreases (at approx. hour 45), the edge cost function is changed to energy-efficiency to save energy during the night hours. During the rest of the simulation (days three and four), HyLERA keeps its blocking probability very low and close to the pure load-balancing SPF, avoiding the blocking peaks experienced by the pure energy-efficient SPF.

It is worth to note that, when working for example in load-balancing mode, HyLERA does not present equal results than the pure load balancing algorithm even if they are actually using the same cost function. This happens since the network has been operated using different routing schemes from the beginning of the simulation, thus creating completely different network states (paths of the connections and free bandwidth on the links) when change in the HyLERA cost function occurs. The same holds for the energy-efficiency modality.

The Figure B.5b shows the total blocking during the simulation. The first day of the simulation obviously presents a similar behaviour to the partial block graph shown in Figure B.5a. The difference between the algorithms can be clearly seen in the second day, where the network is fully loaded and the repetitive traffic pattern starts over. The pure load balancing algorithm maintains a low blocking profile, presenting a very low variation in the values obtained: these are actually the lower bound values of the achievable blocking. The pure energy-aware algorithm notably increases its blocking rate every day until it reaches a stable value on day four, which is substantially higher that the pure load balancing one. HyLERA algorithm has a behaviour similar to the pure load balancing SPF. At the end of day two and for the rest of the days of the simulation, its blocking rates are always within a short range of values from the pure load-balancing ones.

The energy consumption of the three algorithms in the sliding window is reported in Figure B.5c. Here we can see that HyLERA actually starts acting since the very first day of simulation as for the energy consumption, even if the results were not evident as for the blocking in day one.

Since the first day, the energy consumption exhibits a pattern that is repeated until the end of the simulation. As HyLERA starts using the energy-aware function, its energy consumption values are almost identical to the pure energy-aware algorithm and, when the function change occurs, it can be clearly seen that the energy consumption increases up to values close to the ones of the pure load balancing algorithm, but still lower. Note that, as before, HyLERA does not consume the same amount of energy as the pure load balancing algorithm when it is working in load-balancing modality, since the network has been operated in a different way in the close past and is therefore in a different state. When the second function change occurs in day one, HyLERA starts to behave as a pure energy-aware algorithm again, and the energy consumption drastically decreases and the values obtained are very close to the pure energy-aware algorithm.

Finally, we show in Figure B.5d the total energy consumption of the network, taking into account all the connections since the beginning of the simulation. This graphic shows that the HyLERA energy consumption lays almost exactly in between the two pure SPF algorithms, since HyLERA uses during nearly half of the day one cost function and the other function during the other half of the day.

Table B.4 shows the final numerical results of the simulation. Comparing the final blocking of the network for the three different algorithms, we see that the pure energy-aware algorithm has almost twice the blocking percentage of the pure load balancing algorithm, whilst HyLERA (simulation #1) has a slightly higher value than the pure load balancing algorithm very far from the pure energy-aware SPF. About the energy consumption, HyLERA consumes around 40 kWh less (from the 572 kWh in total) than the pure load balancing algorithm maintaining almost the same blocking percentage. As it has been shown, the relation between blocked connections and energy consumption is a trade-off, and if we want to reduce one of them, we have to necessarily increase the other, but HyLERA has achieved a substantial energy reduction at the expense of a very limited increase in the blocking percentage.

Algorithm	**Total Conns**	**Blocked Conns**	**Blocking percentage**	**Consumed energy**
Load Balancing SPF	263,200	1,503	0.5710	572.85
Energy Aware SPF	263,200	2,671	1.0148	497.06
HyLERA Sim. #1	263,200	1,702	0.6466	532.35
HyLERA Sim. #2 (EE)	263,200	2,510	0.9536	513.18
HyLERA Sim. #3 (LB)	263,200	1,652	0.6277	543.99

Table B.4: Final simulation values for the three set of simulations.

Simulation #2: achieving high energy-efficiency (and load-balancing)

The main objective of this simulation set is to show the flexibility of HyLERA to operate in high energy saving mode, leaving only few peak hours in which HyLERA operates balancing the load of the network. Therefore, in order to obtain such a behaviour, the load balancing threshold (t_{high}) and the energy threshold (t_{low}) have been increased (to make it more difficult to switch to load-balancing and more easy to switch back to energy-efficiency).

Figure B.6a shows the blocking probability of the network during the four simulation days (as observed in the sliding window). As in simulation #1, during the first day no appreciable differences in the three algorithms is found. When the second day starts, it can be seen that HyLERA soon switches to energy-awareness and stays in that modality much longer than before, just switching back to load-balancing during few peak hours. As a consequence, HyLERA performs very close to the pure energy aware algorithm since they are actually using the same cost function and the initial conditions of the network were the same (empty network). When the cost function changes to load-balancing, HyLERA continues to obtain close values to the pure energy awareness algorithm for a while, reducing them little by little as the the sliding window is being filled with connections routed with the load balancing modality. This little reduction is due to the tiny contribution of the pure load balancing cost function since it is only being used for 5 hours every day.

Figure B.6b shows the total blocking during the simulation. As it happened on the previous simulation, during the first day the three algorithms have almost the same blocking, being the difference negligible. When the second day starts, and the network gets more congested, since HyLERA is using most of the time the energy-aware function, its behaviour is closer to the pure energy-aware algorithm, but the short range of hours where HyLERA uses the load balancing function positively affects its blocking percentage, obtaining values that are always lower than the pure energy-aware SPF ones.

In Figure B.6c, the evolution of the partial consumption is illustrated. As it happened in the simulation #1, the behaviour is a daily pattern that is repeated since day one. HyLERA starts using the energy-aware cost function, and that is why its consumption starts overlapped with it. It can be noted that, when the change to load-balancing occurs (blue line), an increase in the consumption is recorded, up to intermediate values between the two pure SPF algorithms. When HyLERA switches back to energy-awareness, it quickly decreases its energy consumption to the same values of the pure energy-aware SPF algorithm.

Figure B.6d shows the total energy consumption of the three algorithms. Focusing on HyLERA, it can be seen that its tendency has decreased compared with the first simulation, and its cost is closer to the pure energy-awareness algorithm.

In Table B.4, we can observe that the blocking percentage of HyLERA (simulation #2) is now substantially higher than in the simulation #1 (but still lower than pure energy-aware SPF), and the consumed energy has significantly decreased, saving 60 kWh (from the 572 kWh in total).

This configuration can be used by network operators with relative low traffic who want to save as much energy as possible to reduce their CAPEX expenses as much as possible, while preserving peak hours resource provisioning.

Simulation #3: achieving high load-balancing (and energy-efficiency)

This simulation tries to achieve the opposite behaviour than the simulation #2. In this case, the thresholds have been decreased in order to force HyLERA to use the load balancing cost function during almost all the day time, and thus preserving the connectivity provisioning while not totally discarding energy-efficiency.

Figure B.7a shows the blocking of the algorithms during the sliding window. As in the previous simulations, the behaviour of day one is the same for the three algorithms since the network is still being loaded with connections. Here we can see that the change to load-balancing closely follows the change to energy-efficiency, achieving the required more load-balanced behaviour. During the four days of simulation, HyLERA maintains a performance very close to the pure load-balancing SPF algorithm, slightly changing towards the energy-aware SPF during few night hours.

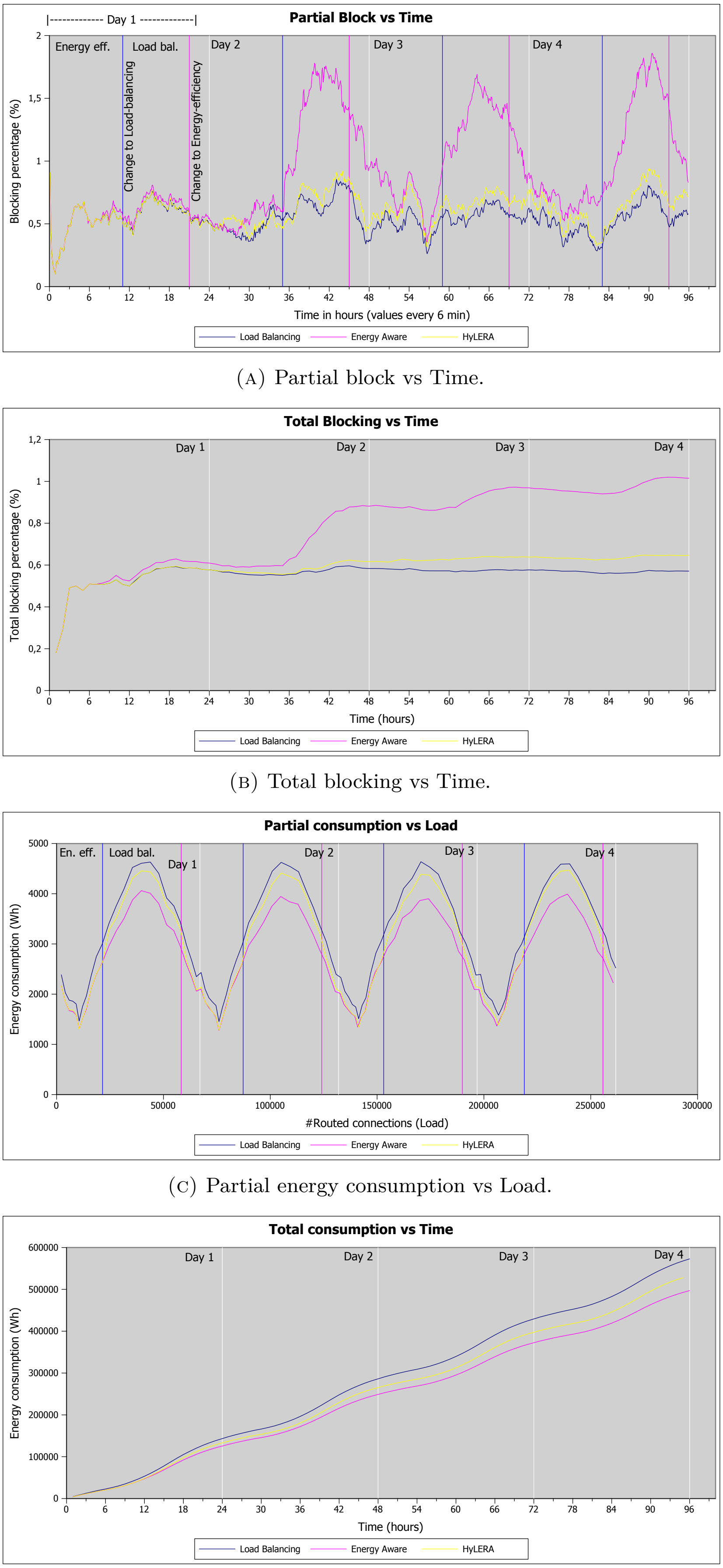
Figure B.7b shows the total blocking of the three algorithms, in which it can be seen how HyLERA performs very well in terms of blocking probability.

Figure B.7c shows the energy consumption of the algorithms during the sliding window. As HyLERA is behaving almost all the day time as the pure load balancing algorithm, their consumptions are very close. Only when HyLERA uses the energy-aware cost function it can be seen how its consumption decreases, but the rest of the time its values are the closest ones with the pure load balancing algorithm among all three simulations.

Figure B.7d shows the total energy consumption. As expected, reducing the blocking percentage makes HyLERA consume more energy, even if its energy consumption stays lower than the pure load-balancing SPF algorithm.

In Table B.4, we can observe that the total energy consumption of HyLERA (simulation #3) is higher than the pure energy-aware SPF but still lower than the pure load-balancing algorithm, saving energy (28 kWh from the total of 572 kWh) while maintaining a total blocking very close to the pure load balancing algorithm.

Considering an average energy cost of 0.12 € per kWh [103, 104], HyLERA saves 443 € per year in Scenario 1 (7% savings with respect to pure LB over a year), 653 € in Scenario 2 (10% savings) and 316 € in Scenario 3 (5% savings). Note that the economic benefits of HyLERA have been calculated within the considered simulation environment, which is an IP/WDM optical network in which every node has its counterpart in the optical domain (i.e. a full optical layer is present and electronic processing only occurs at network edges). This can be considered as the best-case scenario in terms of energy demand, since the use of optical technology introduces significant energy savings. Nevertheless, the HyLERA algorithm can be seamlessly applied to electronic networks in which there are no optical nodes or the optical elements are limited to the core segment. Since the power consumption of electronic devices is as much as 100 times higher than the optical components' [105], the economic benefits of operating HyLERA in such networks easily rises to many thousands of Euro per year. As it has been shown, the relation between blocked connections and energy consumption is a trade-off, and if we want to reduce one of them, we have to necessarily increase the other, but HyLERA achieves a substantial energy reduction at the expense of a very limited increase in the blocking percentage.



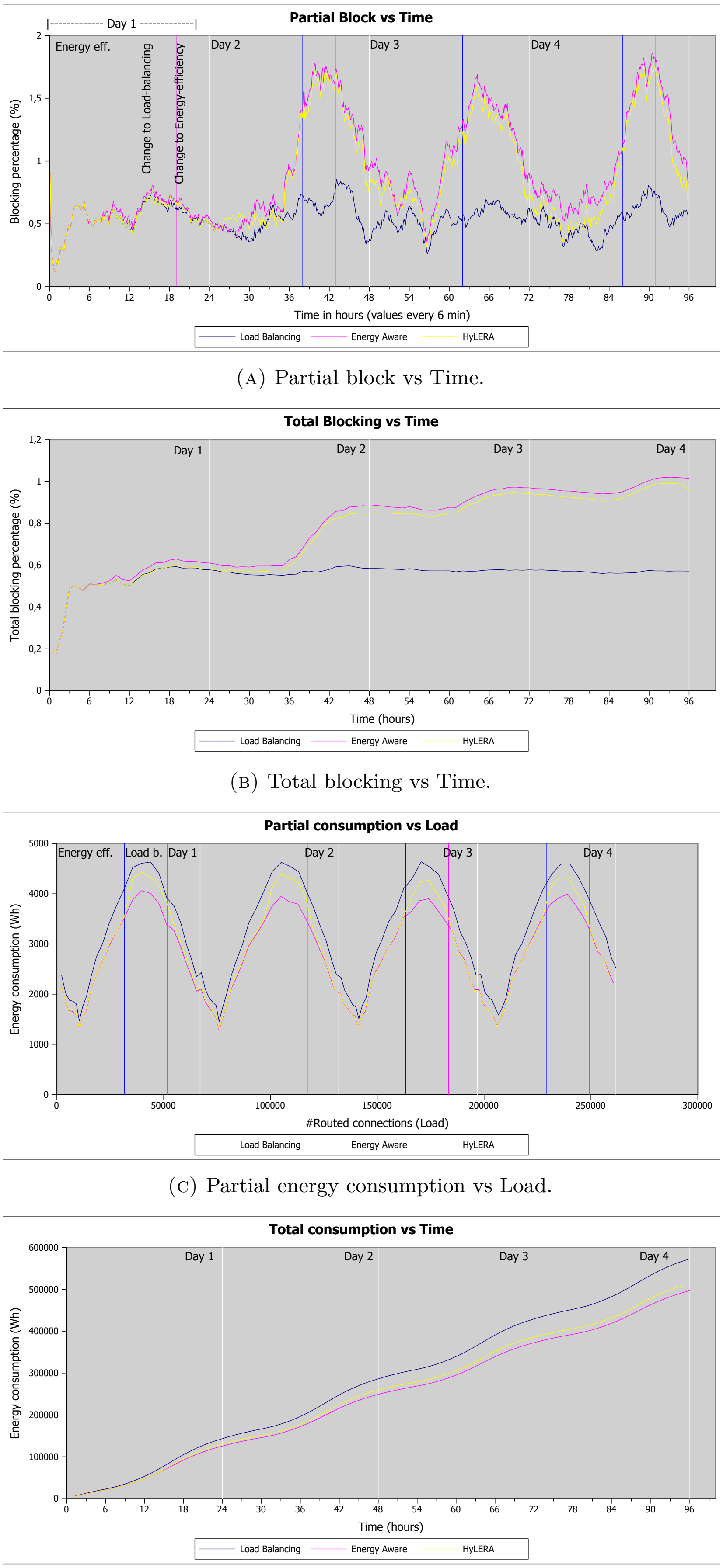
(A) Partial block vs Time.

(B) Total blocking vs Time.

(C) Partial energy consumption vs Load.

(D) Total energy consumption vs Load.

FIGURE B.5: Results for Simulation #1.



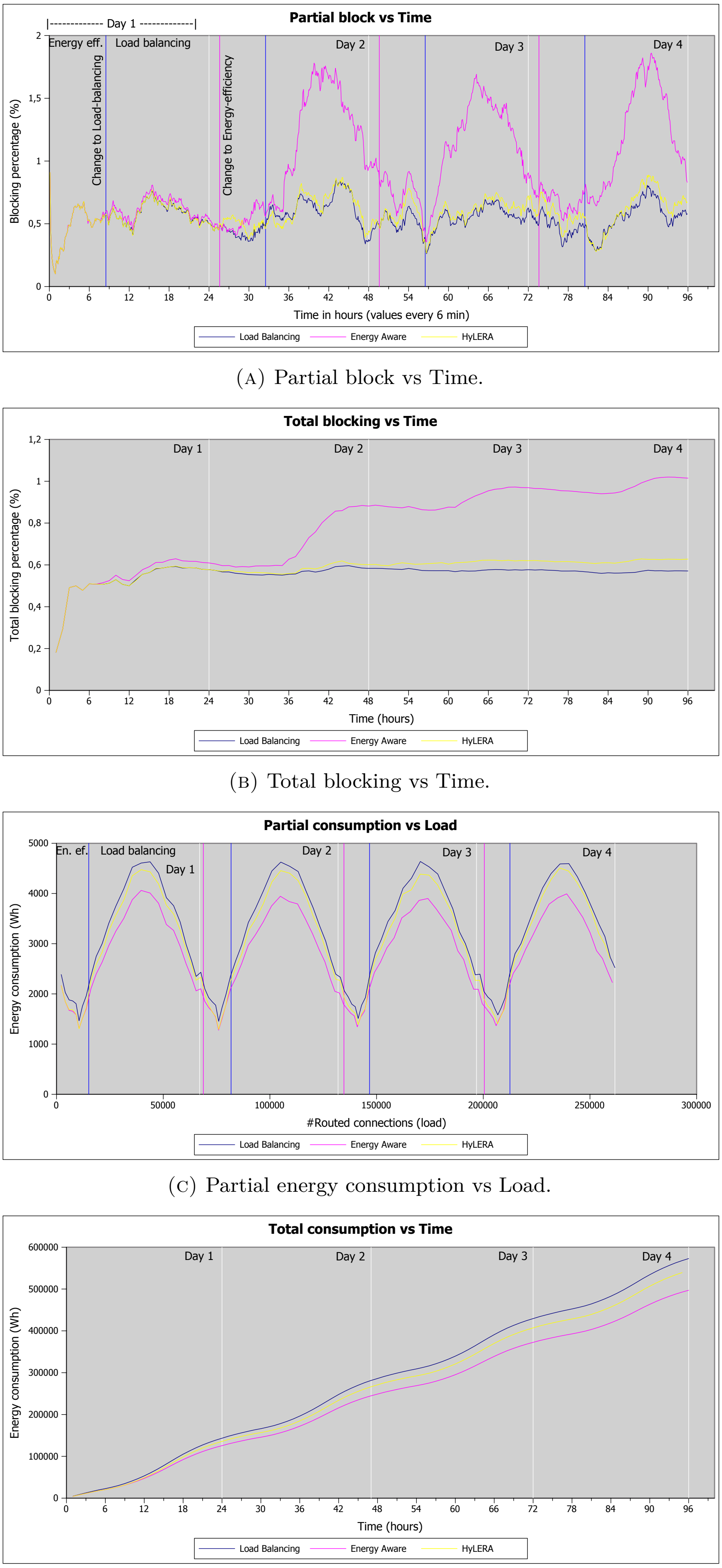
(A) Partial block vs Time.

(B) Total blocking vs Time.

(C) Partial energy consumption vs Load.

(D) Total energy consumption vs Load.

Figure B.6: Results for Simulation #2.



(A) Partial block vs Time.

(B) Total blocking vs Time.

(C) Partial energy consumption vs Load.

(D) Total energy consumption vs Load.

FIGURE B.7: Results for Simulation #3.

Bibliography

[1] Lotfi Belkhir and Ahmed Elmeligi. “Assessing ICT global emissions footprint: Trends to 2040 and recommendations”. In: *Journal of Cleaner Production* 177 (2018), pp. 448 –463. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2017.12.239.

[2] Nathan J. L. Lenssen et al. “Improvements in the GISTEMP Uncertainty Model”. In: *Journal of Geophysical Research: Atmospheres* 124.12 (2019), pp. 6307–6326. DOI: 10.1029/2018JD029522.

[3] *Official Spanish Electric Report*. 2014. URL: http://www.ree.es/sites/default/files/downloadable/avance_informe_sistema_electrico_2014b.pdf.

[4] Commission Staff. “Assessment of the draft National Energy and Climate Plan of Spain”. In: (2019). URL: https://ec.europa.eu/energy/sites/ener/files/documents/es_swd_en.pdf.

[5] *Cisco Visual Networking Index: Forecasts and Trends 2017-2022*. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf.

[6] Sergio Ricciardi et al. “A hybrid load-balancing and energy-aware RWA algorithm for telecommunication networks”. In: *Computer Communications* 77 (2016), pp. 85 –99. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2015.06.010.

[7] *U.S. Energy Information Administration Building Consumption Data*. URL: http://www.eia.gov/tools/faqs/faq.cfm?id=86&t=1.

[8] *Energy Consumption by sector*. 2018. URL: https://www.eia.gov/totalenergy/data/monthly/pdf/sec2.pdf.

[9] *The Energy Performance of Buildings Directive*. 2019. URL: https://ec.europa.eu/energy/sites/ener/files/documents/buildings_performance_factsheet.pdf.

[10] International Telecommunication Union. “Series Y: Global Information Infrastructure, Internet Protocol aspects and Next-Generation Networks”. In: (2012). DOI: 10.1109/TII.2014.2300753.

[11] *That "Internet of Things" thing*. URL: http://www.rfidjournal.com/articles/view?4986.

[12] P. P. Ray. “Towards an Internet of Things based architectural framework for defence”. In: *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. 2015, pp. 411–416. DOI: 10.1109/ICCICCT.2015.7475314.

[13] Oladayo Bello, Sherali Zeadally, and Mohamad Badra. “Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT)”. In: *Ad Hoc Networks* 57 (2017). Special Issue on Internet of Things and Smart Cities security,

privacy and new technologies, pp. 52 –62. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2016.06.010.

[14] L. Catarinucci et al. “An IoT-Aware Architecture for Smart Healthcare Systems”. In: *IEEE Internet of Things Journal* 2.6 (2015), pp. 515–526. ISSN: 2327-4662. DOI: 10.1109/JIOT.2015.2417684.

[15] Zhe Yang et al. “An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare”. In: *Journal of Medical Systems* 40.12 (2016), p. 286. ISSN: 1573-689X. DOI: 10.1007/s10916-016-0644-9.

[16] K. Aziz et al. “Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies”. In: *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*. 2016, pp. 1–7. DOI: 10.1109/ICBDSC.2016.7460394.

[17] L. Li, K. Ota, and M. Dong. “When Weather Matters: IoT-Based Electrical Load Forecasting for Smart Grid”. In: *IEEE Communications Magazine* 55.10 (2017), pp. 46–51. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1700168.

[18] A. Khanna and R. Anand. “IoT based smart parking system”. In: *2016 International Conference on Internet of Things and Applications (IOTA)*. 2016, pp. 266–270. DOI: 10.1109/IOTA.2016.7562735.

[19] D. Minoli, K. Sohraby, and B. Occhiogrosso. “IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems”. In: *IEEE Internet of Things Journal* 4.1 (2017), pp. 269–283. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2647881.

[20] Jalpa Shah and Biswajit Mishra. “Customized IoT Enabled Wireless Sensing and Monitoring Platform for Smart Buildings”. In: *Procedia Technology* 23 (2016). 3rd International Conference on Innovations in Automation and Mechatronics Engineering 2016, ICIAME 2016 05-06 February, 2016, pp. 256 –263. ISSN: 2212-0173. DOI: 10.1016/j.protcy.2016.03.025.

[21] Yuvraj Agarwal et al. “Occupancy-driven Energy Management for Smart Building Automation”. In: *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. BuildSys ’10. Zurich, Switzerland: ACM, 2010, pp. 1–6. ISBN: 978-1-4503-0458-0. DOI: 10.1145/1878431.1878433.

[22] H. Arasteh et al. “Iot-based smart cities: A survey”. In: *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*. 2016, pp. 1–6. DOI: 10.1109/EEEIC.2016.7555867.

[23] Y. Mehmood et al. “Internet-of-Things-Based Smart Cities: Recent Advances and Challenges”. In: *IEEE Communications Magazine* 55.9 (2017), pp. 16–24. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600514.

[24] *Bluetooth Low Energy Whitepaper*. Rev. 1. LitePoint. 2012.

[25] R. Frank et al. “Bluetooth Low Energy: An alternative technology for VANET applications”. In: *Wireless On-demand Network Systems and Services (WONS), 2014 11th Annual Conference on*. 2014, pp. 104–107. DOI: 10.1109/WONS.2014.6814729.

[26] M. Siekkinen et al. "How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4". In: *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*. 2012, pp. 232–237.

[27] J. W. Hui and D. E. Culler. "IPv6 in Low-Power Wireless Networks". In: *Proceedings of the IEEE* 98.11 (2010), pp. 1865–1878. ISSN: 0018-9219. DOI: 10.1109/JPROC.2010.2065791.

[28] Wi-Fi Alliance. *Wi-Fi Alliance introduces low power, long range Wi-Fi HaLow*. https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-low-power-long-range-wi-fi-halow.

[29] N. Sornin et al. *LoRaWAN Specification*. Rev. 1. LoRa Alliance. 2015.

[30] R. Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 7540. Network Working Group, 1997.

[31] R. Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 7540. Network Working Group, 1999.

[32] M. Belshe, R. Peon, and M. Thompson. *Hypertext Transfer Protocol Version 2 (HTTP/2)*. RFC 7540. Internet Engineering Task Force (IETF), 2015.

[33] *CoAP RFC 7252 Constrained Application Protocol*. http://coap.technology.

[34] Z. Shelby, K. Hartke, and C. Bormann. *The Constrained Application Protocol (CoAP)*. RFC 7252. Internet Engineering Task Force (IETF), 2014.

[35] *Information technology: Message Queuing Telemetry Transport (MQTT) v3.1.1*. Standard. https://www.iso.org/standard/69466.html. International Organization for Standardization, June 2016.

[36] A. Barolli, F. Xhafa, and M. Takizawa. "Optimization Problems and Resolution Methods for Node Placement in Wireless Mesh Networks". In: *2011 14th International Conference on Network-Based Information Systems*. 2011, pp. 126–134. DOI: 10.1109/NBiS.2011.28.

[37] T. T. Nguyen et al. "Optimization for the sensor placement problem in 3D environments". In: *2015 IEEE 12th International Conference on Networking, Sensing and Control*. 2015, pp. 327–333.

[38] L. A. Belhaj et al. "Smart-sensor placement optimization under energy objectives". In: *2016 Global Information Infrastructure and Networking Symposium (GIIS)*. 2016, pp. 1–7.

[39] S. Roy and N. Mukherjee. "Integer linear programming formulation of optimal beacon placement problem in WSN". In: *2014 Applications and Innovations in Mobile Computing (AIMoC)*. 2014, pp. 111–117. DOI: 10.1109/AIMOC.2014.6785528.

[40] Antonio Capone et al. "Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach". In: *Computer Communications* 33.10 (2010), pp. 1151 –1161. ISSN: 0140-3664.

[41] D. Djenouri and M. Bagaa. "Energy-Aware Constrained Relay Node Deployment for Sustainable Wireless Sensor Networks". In: *IEEE Transactions on Sustainable Computing* 2.1 (2017), pp. 30–42. ISSN: 2377-3782. DOI: 10.1109/TSUSC.2017.2666844.

[42] Y. Drabu and H. Peyravi. “Gateway Placement with QoS Constraints in Wireless Mesh Networks”. In: *Seventh International Conference on Networking (icn 2008)*. 2008, pp. 46–51. DOI: 10.1109/ICN.2008.89.

[43] Jingzhi Ding, Jianxiao Xu, and Zhifeng Zheng. “Gateway Deployment Optimization in Wireless Mesh Network: A Case Study in China”. In: *2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*. 2009, pp. 300–305. DOI: 10.1109/SOLI.2009.5203949.

[44] B. O. Kahjogh et al. “The impact of critical node elimination on the latency of wireless sensor networks”. In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN 2017)*. 2017, pp. 182–187. DOI: 10.1109/ICUFN.2017.7993771.

[45] H. U. Yildiz et al. “The Impact of Incapacitation of Multiple Critical Sensor Nodes on Wireless Sensor Network Lifetime”. In: *IEEE Wireless Communications Letters* 6.3 (2017), pp. 306–309. ISSN: 2162-2337. DOI: 10.1109/LWC.2017.2679744.

[46] A. Yuksel, E. Uzun, and B. Tavli. “The impact of elimination of the most critical node on Wireless Sensor Network lifetime”. In: *2015 IEEE Sensors Applications Symposium (SAS)*. 2015, pp. 1–5.

[47] H. U. Yildiz, M. Temiz, and B. Tavli. “Impact of Limiting Hop Count on the Lifetime of Wireless Sensor Networks”. In: *IEEE Communications Letters* 19.4 (2015), pp. 569–572. ISSN: 1089-7798.

[48] Yazeed Yasin Ghadi, M.G. Rasul, and M.M.K. Khan. “Design and development of advanced fuzzy logic controllers in smart buildings for institutional buildings in subtropical Queensland”. In: *Renewable and Sustainable Energy Reviews* 54.Supplement C (2016), pp. 738 –744. ISSN: 1364-0321. DOI: 10.1016/j.rser.2015.10.105.

[49] Jang, Hyeonwoo et al. “Design and Implementation of IoT-based HVAC and Lighting System for Energy Saving”. In: *MATEC Web Conf.* 260 (2019), p. 02012. DOI: 10.1051/matecconf/201926002012.

[50] D. Minoli, K. Sohraby, and B. Occhiogrosso. “IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems”. In: *IEEE Internet of Things Journal* 4.1 (2017), pp. 269–283. ISSN: 2372-2541. DOI: 10.1109/JIOT.2017.2647881.

[51] Pervez Hameed Shaikh et al. “Intelligent multi-objective optimization for building energy and comfort management”. In: *Journal of King Saud University - Engineering Sciences* (2016). ISSN: 1018-3639. DOI: 10.1016/j.jksues.2016.03.001.

[52] ANSI/ASHRAE. *Thermal Environmental Conditions for Human Occupancy*. 2004.

[53] Anuj Kumar, I P Singh, and S K Sud. “AN APPROACH TOWARDS DEVELOPMENT OF PMV BASED THERMAL COMFORT SMART SENSOR”. In: *International Journal on Smart Sensing and Intelligent Systems* 3 (2010), pp. 621–642. DOI: 10.21307/ijssis-2017-412.

[54] Mohd Izani Mohamed Rawi and Adnan Al-Anbuky. “Development of Intelligent Wireless Sensor Networks for Human Comfort Index Measurement”. In: *Procedia Computer Science* 5.Supplement C (2011). The 2nd International Conference on Ambient Systems,

Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011), pp. 232 –239. ISSN: 1877-0509.

[55] A. Kumar and G. P. Hancke. “An Energy-Efficient Smart Comfort Sensing System Based on the IEEE 1451 Standard for Green Buildings”. In: *IEEE Sensors Journal* 14.12 (2014), pp. 4245–4252. ISSN: 1530-437X. DOI: 10.1109/JSEN.2014.2356651.

[56] National Optical Astronomy Observatory. *Recommended Light Levels*. 2015.

[57] N. Wang, F. Fang, and M. Feng. “Multi-objective optimal analysis of comfort and energy management for intelligent buildings”. In: *The 26th Chinese Control and Decision Conference (2014 CCDC)*. 2014, pp. 2783–2788. DOI: 10.1109/CCDC.2014.6852646.

[58] J. A. Pinzon et al. “An MILP model for optimal management of energy consumption and comfort in smart buildings”. In: *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 2017, pp. 1–5. DOI: 10.1109/ISGT.2017.8085956.

[59] P. Desai, A. Sheth, and P. Anantharam. “Semantic Gateway as a Service Architecture for IoT Interoperability”. In: *2015 IEEE International Conference on Mobile Services*. 2015, pp. 313–319. DOI: 10.1109/MobServ.2015.51.

[60] R. Mahmoud et al. “Internet of things (IoT) security: Current status, challenges and prospective measures”. In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015, pp. 336–341. DOI: 10.1109/ICITST.2015.7412116.

[61] Miao Wu et al. “Research on the architecture of Internet of Things”. In: *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings* 5 (2010), pp. 484–487. DOI: 10.1109/ICACTE.2010.5579493.

[62] Luigi Atzori et al. “The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization”. In: *Computer Networks* 56.16 (2012), pp. 3594 –3608. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2012.07.010.

[63] Zhihong Yang et al. “Study and application on the architecture and key technologies for IOT”. In: *2011 International Conference on Multimedia Technology*. 2011, pp. 747–751. DOI: 10.1109/ICMT.2011.6002149.

[64] Sotiris Papantoniou, Stefano Mangili, and Ivan Mangialenti. “Using Intelligent Building Energy Management System for the Integration of Several Systems to one Overall Monitoring and Management System”. In: *Energy Procedia* 111 (2017). 8th International Conference on Sustainability in Energy and Buildings, SEB-16, 11-13 September 2016, Turin, Italy, pp. 639 –647. ISSN: 1876-6102. DOI: 10.1016/j.egypro.2017.03.226.

[65] Vlasios Tsiatsis et al. “The SENSEI real world internet architecture”. In: *Towards the Future Internet: Emerging Trends from European Research* (2010), pp. 247–256. DOI: 10.3233/978-1-60750-539-6-247.

[66] Martin Bauer et al. “Project Deliverable D1.5 – Final Architectural Reference Model for IoT”. In: (2013). http://www.iot-a.eu, pp. 53–59.

[67] R. Khan et al. “Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges”. In: *2012 10th International Conference on Frontiers of Information Technology*. 2012, pp. 257–260. DOI: 10.1109/FIT.2012.53.

[68] 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. 2010, pp. V5–484–V5–487. DOI: 10.1109/ICACTE.2010.5579493.

[69] I. Yaqoob et al. “Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges”. In: *IEEE Wireless Communications* 24.3 (2017), pp. 10–16. ISSN: 1558-0687. DOI: 10.1109/MWC.2017.1600421.

[70] L. D. Xu, W. He, and S. Li. “Internet of Things in Industries: A Survey”. In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. ISSN: 1941-0050. DOI: 10.1109/TII.2014.2300753.

[71] Hui Suo et al. “Security in the Internet of Things: A Review”. In: *Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012* 3 (Mar. 2012). DOI: 10.1109/ICCSEE.2012.373.

[72] Jaehak Yu et al. “IoT as a applications: cloud-based building management systems for the internet of things”. In: *Multimedia Tools and Applications* 75.22 (2016), pp. 14583–14596. ISSN: 1573-7721. DOI: 10.1007/s11042-015-2785-0.

[73] M. A. Chaqfeh and N. Mohamed. “Challenges in middleware solutions for the internet of things”. In: *2012 International Conference on Collaboration Technologies and Systems (CTS)*. 2012, pp. 21–26. DOI: 10.1109/CTS.2012.6261022.

[74] N. Mohamed, S. Lazarova-Molnar, and J. Al-Jaroodi. “CE-BEMS: A cloud-enabled building energy management system”. In: *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*. 2016, pp. 1–6. DOI: 10.1109/ICBDSC.2016.7460393.

[75] Ala Al-Fuqaha et al. “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. In: *IEEE Communications Surveys and Tutorials* 17.4 (2015), pp. 2347–2376. DOI: 10.1109/COMST.2015.2444095.

[76] Luka Milić and Leonardo Jelenković. “A novel versatile architecture for Internet of Things”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings* (2015), pp. 1026–1031. DOI: 10.1109/MIPRO.2015.7160426.

[77] P. Spiess et al. “SOA-Based Integration of the Internet of Things in Enterprise Services”. In: *2009 IEEE International Conference on Web Services*. 2009, pp. 968–975. DOI: 10.1109/ICWS.2009.98.

[78] 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 and Electrical Engineering* 65 (2018), pp. 222 –235. ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2017.09.001.

[79] M. Manic et al. “Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings”. In: *IEEE Industrial Electronics Magazine* 10.1 (2016), pp. 25–39. ISSN: 1941-0115. DOI: 10.1109/MIE.2015.2513749.

[80] *AS-XM1000 Mote Module*. URL: http://www.advanticsys.com/shop/asxm1000-p-24.html.

[81] Álvaro Villalba et al. "servIoTicy and iServe: A Scalable Platform for Mining the IoT". In: *Procedia Computer Science* 52 (2015). The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015), pp. 1022 –1027. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.05.097.

[82] *Barcelona Supercomputing Center*. http://www.bsc.es.

[83] *COMPOSE: Collaborative Open Market to Place Objects at your Service*. http://www.compose-project.eu.

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

[85] Javier Vales-Alonso et al. "Performance evaluation of MAC transmission power control in wireless sensor networks". In: *Computer Networks* 51.6 (2007), pp. 1483 –1498. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2006.08.001.

[86] Thomas A Feo and Mauricio G.C Resende. "A probabilistic heuristic for a computationally difficult set covering problem". In: *Operations Research Letters* 8.2 (1989), pp. 67 –71. ISSN: 0167-6377. DOI: 10.1016/0167-6377(89)90002-3.

[87] M. Ficco, F. Palmieri, and A. Castiglione. "Hybrid indoor and outdoor location services for new generation mobile terminals". In: *Personal and Ubiquitous Computing* 18.2 (2014), pp. 271–285. DOI: 10.1007/s00779-013-0644-4.

[88] Rabee M. Reffat and Edward L. Harkness. "Environmental Comfort Criteria: Weighting and Integration". In: *Journal of Performance of Constructed Facilities* 15 (3 2001). DOI: 10.1061/(ASCE)0887-3828(2001)15:3(104).

[89] BONE project. *WP 21 TP Green Optical Networks, D21.2b Report on Y1 and updated plan for activities*. 2009.

[90] *Living Planet Report 2010, The biennial report, WWF, Global Footprint Network, Zoological Society of London*. 2010.

[91] The Climate Group. *SMART 2020: Enabling the low carbon economy in the information age*. 2008.

[92] B. St Arnaud. *ICT and Global Warming: Opportunities for Innovation and Economic Growth*.

[93] J. Chabarek et al. "Power Awareness in Network Design and Routing". In: *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. 2008, pp. 457–465. DOI: 10.1109/INFOCOM.2008.93.

[94] W. Vereecken et al. "Overall ICT footprint and green communication technologies". In: *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on*. 2010, pp. 1–6. DOI: 10.1109/ISCCSP.2010.5463327.

[95] Sergio Ricciardi et al. “Towards an energy-aware Internet: modeling a cross-layer optimization approach”. In: *Telecommunication Systems* (2011), pp. 1–22. ISSN: 1018-4864. DOI: 10.1007/s11235-011-9645-7.

[96] A. Muhammad et al. “Energy-Efficient WDM Network Planning with Dedicated Protection Resources in Sleep Mode”. In: *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*. 2010, pp. 1 –5. DOI: 10.1109/GLOCOM.2010.5683205.

[97] J. Moy. *OSPF version 2*. RFC 2178. RFC Editor, 1998. URL: https://tools.ietf.org/html/rfc2328.

[98] Jeff Doyle and Jennifer X. Carroll. *Routing TCP/IP, Volume 1 (2nd Edition)*. Cisco Press, 2005. ISBN: 1587052024.

[99] R. Ramamurthy and B. Mukherjee. “Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks”. In: *Networking, IEEE/ACM Transactions on* 10.3 (2002), pp. 351 –367. ISSN: 1063-6692.

[100] Sergio Ricciardi et al. “An energy-aware dynamic RWA framework for next-generation wavelength-routed networks”. In: *Computer Networks* 56.10 (2012), pp. 2420 –2442. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2012.03.016.

[101] F. Palmieri, U. Fiore, and S. Ricciardi. “SimulNet: a wavelength-routed optical network simulation framework”. In: *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*. 2009, pp. 281 –286. DOI: 10.1109/ISCC.2009.5202259.

[102] L. Chiaraviglio, M. Mellia, and F. Neri. “Energy-Aware Backbone Networks: A Case Study”. In: *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*. 2009, pp. 1–5. DOI: 10.1109/ICCW.2009.5208038.

[103] *Europe electricity prices*. 2015. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php/Energy_price_statistics.

[104] *United states energy information administration, state electricity prices*. 2014. URL: http://www.eia.doe.gov/neic/rankings/stateelectricityprice.htm.

[105] Sergio Ricciardi et al. “Chapter 19 - Towards Energy-Oriented Telecommunication Networks”. In: *Handbook of Green Information and Communication Systems*. Ed. by Mohammad S. Obaidat, Alagan Anpalagan, and Isaac Woungang. Academic Press, 2013, pp. 491 –512. ISBN: 978-0-12-415844-3. DOI: 10.1016/B978-0-12-415844-3.00019-X.