DOCTORAL THESIS OF

ÉCOLE NORMALE
SUPÉRIEURE DE RENNES
COMUE UNIVERSITÉ BRETAGNE LOIRE

DOCTORAL SCHOOL N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Speciality : Computer Science

By

Loic GUEGAN

Scalable end-to-end models for the time and energy performance of Fog infrastructures

ACKNOWLEDGEMENT

In the first place, I would like to thank my supervisors Anne-Cécile Orgerie and Martin Quinson. They taught me the art of research in Computer Science. During my three years of PhD, they were always present to guide and support me. They were able to quickly identify my personality and then adjust their pedagogical skills in accordance. Therefore, for all of these reasons I will always be grateful to them.

I am grateful to all the jury members for reading my manuscript and for being part of my PhD defense. In addition, I would like to thank the reviewers for their constructive comments that allow me to greatly improve my manuscript.

I would like to thank all the Myriads team members for hosting me during my PhD. I spent very nice working hours in your company.

I also would like to thank The Anh Pham for being part of PhD journey and sharing his joy and Vietnamese culture with me. I would like to thank Ali Jawad Fahs for accompanying me during my PhD and sharing very nice coffee breaks. By the way, thank you for for the towel. I also want to thank Mehdi Belkhiria and Aymen Jlassi for their company and their support.

I am grateful to Onesphore Ndayishimiye for all advices and for the nice conversations that we had together. I want you to know that I have a lot of respect for you and for all you have accomplished.

Then, I am also grateful to all the people who give me the opportunity to discover the research during my master thesis, namely Sylvain Bonhommeau, Nour Mohammad Murad and Richard Lorion. They trusted me on a research subject with a pure engineering background and we achieve very nice work together.

How to not think about my family, my first thanks are for my fantastic parents, my father Bruno Guegan and my mother Sylviane Sinacouty. They always encourage me to pursue my studies as far as I can and they always support me as much as they could. I also would like

3

to thank my sister for her support and the nice moments that we spend together. I am also particularly grateful to my grandmother Luciana Sinacouty, she is an amazing woman and even if she is far away, I know that I can rely on her. I also would like to thank my parents-in-law for the very nice moments that we spend together. Finally, I am grateful of my girlfriend Anais Clayton which has always been present with me during my three years of PhD. She encourages me and support me in all the stressful moments.

And last but not least, I would like to thank all the persons that I forgot to mention.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

11

ACRONYMS

ALS Application-Level Simulator

AP Access Point

ARE Average Relative Error

CDN Content Delivery Network

CR Coding Rate

DCF Distributed Coordination Function

DES Discrete-Event Simulator

DSL Domain Specific Language

FLS Flow-Level Simulator

HIL Hardware In the Loop

HPC High Performance Computing

ICT Information and Telecommunication Technologies

IoT Internet Of Things

ISP Internet Service Provider

LMM Linear MaxMin

MCS Modulation Coding Scheme

MPDU MAC Protocol Data Unit

MSDU MAC Service Data Unit

MTU Maximum Transmission Unit

OSI Open Systems Interconnection

PLS Packet-Level Simulator

PUE Power Usage Effectiveness

Rx Reception

STA Station

TCP Transmission Control Protocol

Tx Transmission

UDP User Datagram Protocol

VM Virtual Machine

WLAN Wireless Local Area Network

WSN Wireless Sensors Network

FRENCH SUMMARY

Contexte

L'informatique géo-distribuée (Fog Computing) est un néologisme attribué à la migration des ressources de calcul et de stockage du nuage (Cloud) vers les utilisateurs. L'objectif de cette migration des ressources est de permettre une réduction des temps de réponse et une augmentation de la bande passante des terminaux des utilisateurs. Cela permet donc de traiter l'information proche de l'utilisateur contrairement au nuage. De plus, cela réduit le nombre de communication entre les terminaux et le nuage, contribuant ainsi la réduction de la charge globale du réseau.

En effet, l'évolution des usages de l'Internet contribue à l'augmentation du nombre de terminaux connectés à Internet mais également à l'augmentation du nombre de communication. De plus, cette tendance tend à s'accélérer avec le développement de l'Internet des objets. Ce paradigme consiste à connecter des objets à Internet par le biais de différentes techniques de communication. Le but étant d'étendre les fonctionnalités primaires des objets en les rendant accessibles sur Internet. Parallèlement à cela, la nature même du trafic est elle aussi en constante évolution avec le développement de service tel que le streaming, utilisant d'énorme quantité de bandes passantes. Ainsi, l'informatique géo-distribuée permet de limiter les effets de cette croissance et d'augmenter les performances du réseau Internet.

De plus, cette révolution des usages de l'Internet a un impact majeur sur sa consommation en énergie. De par le fait que la consommation en énergie du réseau est liée à sa taille et à son taux d'usage, nous faisons face à une augmentation significative de la consommation en énergie des réseaux Internet. Ainsi, en 2010, entre 1.1% et 1.5% de l'énergie consommée par l'Homme est dédiée uniquement aux centres de données [1]. De plus, la gestion de l'énergie consommée par la quantité astronomique d'objets connectés à Internet est un sujet complexe [2]. L'informatique géo-distribuée serait en mesure de réduire cette consommation en énergie des réseaux mais des études approfondies doivent être menées en ce sens afin d'en déterminer la proportion.

Afin d'étudier l'Internet, les scientifiques le subdivisent en composants élémentaires appelés plateformes. Ces plateformes sont composées d'un ensemble d'ordinateurs reliés entre eux à

l'aide de diverses techniques de communication. Ces techniques de communication peuvent être filaires ou sans fil et assurent l'intégrité des données transférées. La communication entre ordinateurs a permis le développement de plusieurs domaines de recherche tels que les technologies de communication, les protocoles de communication, les services en ligne et la consommation d'énergie des réseaux. Ainsi, afin d'étudier les réseaux et faire progresser ces axes de recherche, les scientifiques ont besoin de réaliser des expérimentations sur des plateformes réseaux. Ces expériences peuvent être menées sur des bancs de tests, à savoir des plateformes réseaux dédiées à l'expérimentation. Ce type d'expérimentation a l'avantage de fournir des résultats réels précis.

Cependant les plateformes réseaux réelles sont larges-échelle et cela augmente les difficultés à déployer des bancs de tests. En effet, le déploiement et la configuration des noeuds à large-échelle est beaucoup plus compliqué et a un coût en temps et en argent. Les communications sont également plus difficiles à monitorer et la nature imprévisible des grandes plateformes réseaux rend la reproductibilité des expériences plus complexe. De plus, ces plateformes réseaux sont souvent interdépendantes et communiquent entre-elles, particulièrement sur les infrastructures de type informatique géo-distribuée. Ainsi, une simple étude du réseau de bout-en-bout requiert l'usage de plusieurs plateformes telles que les objets connectés, l'informatique géo-distribuée, le réseaux des fournisseurs d'accès à Internet et le nuage.

La simulation est un processus expérimental largement utilisé dans la recherche. Elle est également utilisée lors de l'étude des réseaux. Cette technique consiste à utiliser la capacité de calcul des ordinateurs afin de prédire les états futurs d'une plateforme réseau en fonction de conditions initiales. Ces états du réseaux peuvent représenter divers métriques telles que la consommation d'énergie, le temps etc.
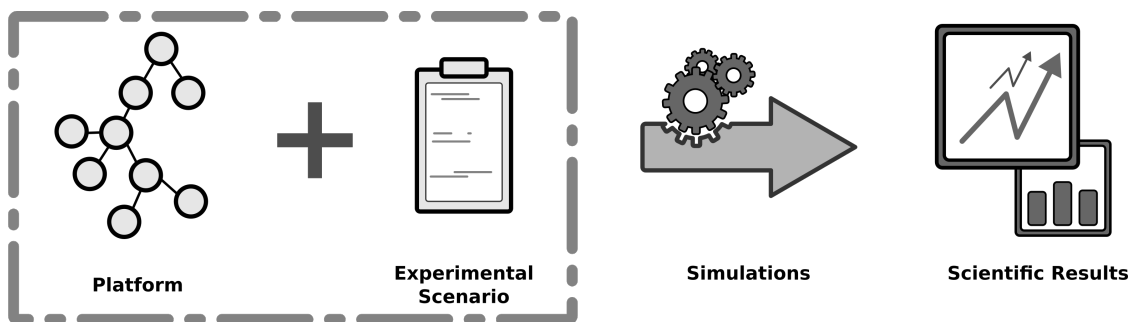
Figure 1: Processus expérimental de la simulation.

Les simulateurs réseaux sont composés de différents modèles en charge de prédire les différents états de la plateforme simulée. Comme présenté sur la Figure 1, un simulateur réseau

requiert deux entrées: une plateforme et un scénario expérimental. Ainsi, le simulateur est en charge d'exécuter le scénario expérimental sur la plateforme réseau en fonction de ces différents modèles et il produit des résultats scientifiques qui peuvent par la suite être étudiés. La précision des résultats d'un simulateur réseau est donc intimement liée à la précision de ces modèles.

Objectifs de recherche et problématique

La simulation offre de nombreux bénéfices en comparaison aux expérimentations réelles. Premièrement, les scientifiques ne sont pas limités par les plateformes physiques car lors de simulations réseaux, elles sont virtuelles. De façon similaire, comme aucune plateforme réelle n'est déployée, les scientifiques économisent du temps, de l'argent et potentiellement de l'énergie. De plus, comme les simulateurs réseaux sont déterministes, les expériences peuvent être reproduites facilitant ainsi les études. Ainsi, les simulateurs réseaux sont une bonne alternative aux expérimentations réelles pour l'étude de l'Internet de bout-en-bout.

Est-il possible d'étudier la consommation d'énergie du réseau de bout-en-bout d'une platforme large-échelle à l'aide de la simulation ?

L'utilisation de la simulation réseau pour l'étude de leur consommation en énergie peut sembler être une idée attrayante. Cependant, les plateformes modernes étant larges-échelle, la simulation réseau pose des problèmes de scalabilité au niveau des performances sur les ordinateurs actuels. En effet, la plupart des simulateurs existants ne sont pas capables de simuler plus d'une centaine de noeuds. En ce qui concerne les simulateurs qui en sont capables, aucun n'est suffisamment versatile afin de simuler le réseau Internet de bout-en-bout. En effet, cela implique d'être en mesure de simuler les objets connectés, l'informatique géo-distribuée, le réseau des fournisseurs d'accès à Internet et le nuage. En conséquent, l'état de l'art actuel en lien avec la simulation réseau, ne fournit pas de solution nécessaire à l'étude de la consommation d'énergie du réseau Internet de bout-en-bout au sein d'un unique simulateur.

Contribution

Ce manuscrit de thèse propose différents modèles de simulation afin de permettre l'étude de la consommation d'énergie des réseaux de bout-en-bout au sein d'un même simulateur. Les contributions de cette thèse sont:

1. Une étude préliminaire de la consommation d'énergie du réseau de bout-en-bout à l'aide des outils actuels d'expérimentation afin de motiver le développement de nos modèles.

2. Un modèle de consommation de l'énergie des réseaux filaires adapté aux simulations larges-échelles.

3. Un modèle de simulation Wi-Fi pour l'étude de réseaux larges-échelle servant de base à une étude de la consommation d'énergie du Wi-Fi.

Publications

- **Loic Guegan**, Anne-Cécile Orgerie. Estimating the end-to-end energy consumption of low-bandwidth IoT applications for WiFi devices. CloudCom 2019 - 11th IEEE International Conference on Cloud Computing Technology and Science, Dec 2019, Sydney, Australia.

- **Loic Guegan**, Betsegaw Amersho, Anne-Cécile Orgerie, Martin Quinson. A Large-Scale Wired Network Energy Model for Flow-Level Simulations. AINA 2019 - 33rd International Conference on Advanced Information Networking and Applications, Mar 2019, Matsue, Japan. pp.1047-1058.

INTRODUCTION

1.1 Context

The Fog computing is a neologism employed to describe the migration of Cloud computing resources to the edge of the network. The goal is to alleviate the load of futur network platforms and to reduce the edge devices response time. Indeed, by bringing computing resources to the edge of the network, data can be processed closer to the end user. This can lead to smaller network communications between Fog nodes and the Cloud, fewer computations in the Cloud and lower latencies. Such a Fog infrastructure is required since future network platforms are growing along with network usages.

In fact, the Internet is continuously evolving. A particular aspect of this evolution is related to its size. Getting access to Internet is now easy and the development of online services has lead to a new generation of large-scale networks. Today's networks can now easily reach thousands of nodes [3] and involve many communications [4]. This trend is powered by the arrival of the IoT which consists in connecting physical objects to the network by mean of numerous technologies involving sensors, wireless communications, network protocols, etc. Therefore the data generated by the objects become available on the Internet. However, the tremendous amount of data generated by these objects will lead to a significant increase of network communications and Cloud data processing. Moreover, the nature of the network traffic is also evolving and the development of high bandwidth applications such as streaming has an unprecedented impact on the Internet network load. Consequently, Fog computing was introduced to mitigate all these effects. Still, these changes of the Internet usage has a non negligible impact on the research domain.

The network energy consumption is a major domain which is directly impacted by this network revolution. Indeed, since the network energy consumption is tied to its size and load, we are facing to an increase of the energy consumed by common networks. As an example, about 1.1% to 1.5% of the energy consumed by the entire world is dedicated solely to Data Centers [1] in 2010. Similarly, sustaining the energy consumed by billion of devices from the IoT

is a critical challenge [2]. Fog computing may help to reduce this energy consumption [5] but studies has to be conduct to quantify in which proportion. Consequently, scientists are working on it, striving to reduce the energy consumed by the Internet.

To study the Internet, scientists usually subdivide it into sub components called network platforms. These network platforms are groups of computers or devices called nodes connected together by mean of various communications technologies. These technologies could ether be wired or wireless and they ensure a resilient information transfer between the nodes that satisfy the performance requirements. Making computers and devices communicating together opened the door to many research domains such as network communication technologies, network protocol, network services, and energy consumption. To conduct research in these domains, scientists has to carry out experiments on network platforms. These experiments can be made on testbeds which are dedicated network platforms used as a research environment. This type of experimentation has the advantage to provide real measurements for accurate research.

However, modern platforms are large-scale. This drastically increase the difficulties to conduct experimentations on testbeds. Deploying nodes becomes time consuming and expensive. Communications are more complicated to track and the natural unpredictability of a large testbeds due to its current state make the experimentations hard to reproduce. On top of that, network platforms are often dependent on each other since they are communicating together specially on Fog infrastructures. Thus, a typical end-to-end network will involved the IoT network, the Fog nodes, the Internet Service Provider (ISP) network and the Cloud.

Simulation is an experimental process widely in engineering and research. It is also used to study networks by computer scientists. This technique consists in using the computational power of computers to predict future states of a network given some initial conditions. The network state could be represented by many characteristics such as time, energy consumption etc.
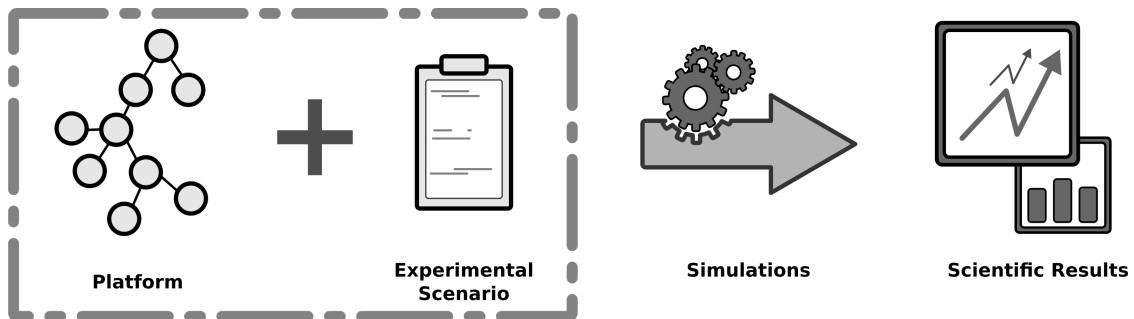
Figure 1.1: Simulation experimental process.

Network simulators are composed of several models which are used to predict the final state of the simulated network. As presented on Figure 1.1 a network simulator require two inputs: a network platform and an experimental scenario. Then, the simulator executes the experimental scenario on the network platform according to its network models and produces scientific results which can be analyzed. Thus, the results accuracy of a simulator is entirely dependent of the models' quality.

1.2 Research problem and goal

Simulation offers great benefits compare to real world experiments. First, scientists are not limited by the physical platform since simulators use a virtual platform. Similarly, since no real platform is deployed, scientists save time, money and potentially energy. Another great feature of network simulation is reproducibility. Since simulators are deterministic, experimentations can be reproduced multiple time which facilitates the studies. Thereby, network simulations might be a alternative to testbeds to study the energy consumption of the network from end-to-end.

Is it possible to study a large-scale end-to-end network energy consumption using network simulations ?

Using simulation to study the energy consumption of large-scale end-to-end network is an appealing idea. Since today's network energy consumption studies require to simulate multiple large-scale platforms, this push current computers hardware to their limits. In fact, most of current network simulators cannot handle more than hundred of nodes. Regarding the scalable ones, to the best of our knowledge, none of them are versatile enough to simulate networks from end-to-end. This involves the edge devices such as IoT, Fog nodes, the Internet Service Provider network and the Cloud. Consequently, in current state of the art related to network simulations, there is no solution available to study the end-to-end network energy consumption with a single network simulator.

1.3   Contribution
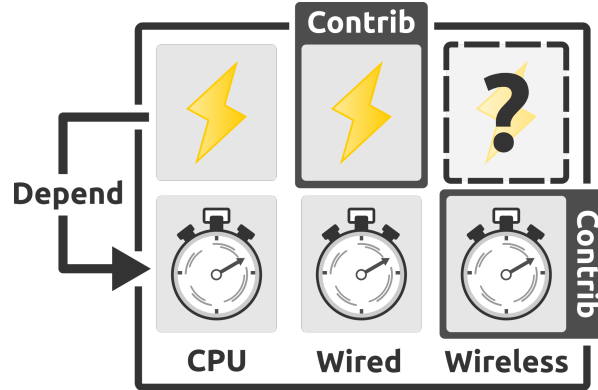
Figure 1.2: Simulation models required for end-to-end network energy consumption. The contributions of this thesis are labeled «Contrib ».

In this thesis, we propose different simulation models to enable the study of the end-to-end network energy consumption with a single simulator. A representation of the required models are depicted in Figure 1.2. This figure shows that two models are required for predicting the energy consumption of a task. First, a performance model which predict the task duration. Second, an energy model which predicts its energy consumption. Hence, the contributions of this thesis are the following:

1. A preliminary end-to-end network energy consumption study using the available experimentation tools, motivating the development of scalable models.

2. A scalable wired network energy model to study the energy consumed by large-scale platforms.

3. A scalable Wi-Fi network model towards the study of large-scale Wi-Fi network energy consumption.

1.4   Organization of the document

The reminder of this thesis in organized as follows. In Chapter 2, we present the state of the art on network simulations. We start by presenting the existing large-scale platforms that we want to study with detailed applications. Then, an analysis of the challenges raised by these

large-scale platforms is addressed. Next, we present the different simulation models used to study them. Then, we introduce the existing network simulators that can be used to study these large-scale platforms. In Chapter 3, we present a motivating study on the end-to-end network energy consumption. We highlight the limits of previously available solutions while proposing future directions towards energy efficient low-bandwidth end-to-end applications. In Chapter 4, we propose a scalable wired network energy model that aims at studying large-scale platform network energy consumption. In this work, we are adapting current wired energy model to coarse-grained network simulator to benefit from their scalability property. In Chapter 5, we propose the first scalable Wi-Fi communication model to estimate Wi-Fi communication durations on large-scale platform. We achieve this work by modeling the Wi-Fi bandwidth sharing allocation mechanism at a coarse-graine level. Finally, we conclude in Chapter 6 and provide future research directions.

STATE OF THE ART

Contents

2.1 Overview of Today's Network Platforms

According to the Internet traffic forecasts [4, 6], the world network load is continuously increasing along with the number of devices connected to the Internet.

These global Internet connection analysis show that the number of device and network communication are increasing about 4% faster than the population size.

This trend can be explained by a raise of the number of connected devices mainly driven by machine to machine communications. They are expected to represent half of the total amount of devices connected to the Internet by 2023. These devices, or machines, can be connected together on a common network to form a platform. Consequently, current platforms are getting larger and complex.

This section is organized as follow. Section 2.1.1 present an overview of common large scale network platforms to characterize them. Then Section 2.1.2 introduce use cases related to these platforms. Finally, Section 2.1.3 present the several research challenges raised by large-scale and heterogenous platforms.

2.1.1   Major Platforms Taxonomy

From the beginning of the Internet, Information and Telecommunication Technologies (ICT) platforms has been the support of communication and the processing entity of the human generated data. Recent studies have shown a raise of the global Internet traffic with more and more communicating and computing intensive services [6]. In response, ICT platforms are growing and evolving proportionally to this demand from end-to-end starting from the Cloud up to the end user. In this section we thus propose to characterize these platforms in terms of their distance from the end user and in therms of their hardware heterogeneity.

Starting by the farthest platform from the end user, data centers were first introduced to address the raise of the network services demand. It consists of thousands of servers grouped together to improve computational and data storage capabilities. Data centers are mainly composed of homogeneous hardware with defined standards [7] and that servers are connected together by hundred of network devices. Since data centers are very large platforms, studying them is challenging. This is why, experimental testbeds were developed to solve this issue. As an example Grid'5000 [8] is a large-scale experimentation testbed located in France. Currently, Grid'5000 composed of 35 clusters spread over 8 different sites. Each cluster has its own hardware specification. This type of platform is used for scientific research related to Cloud, High Performance Computing (HPC), etc. Other large-scale experimentation testbeds have also been developed such as Magellan [9] and CloudLab [10]. These tools simplify the study of these platforms which still difficult to study.

Between far and near platforms lay ISP platforms. ISP connect together the different computers and devices from various regions of the world [11]. ISP platforms are mainly composed network devices such as routers and switches. The different ISP are organized in tiers architecture. Tier 1 ISP form the backbone and are in charge of connecting a given country to remote continents. Tier 2 and tier 3 have smaller network infrastructures and are usually connected to tier 1 ISP. Thus, ISP are very large networks owned by several parties which contains heterogeneous hardware such as routers, switches and servers. Consequently, studying such a large and heterogeneous platform is difficult.

While data centers and ISP networks are large-scale platforms separated from the end
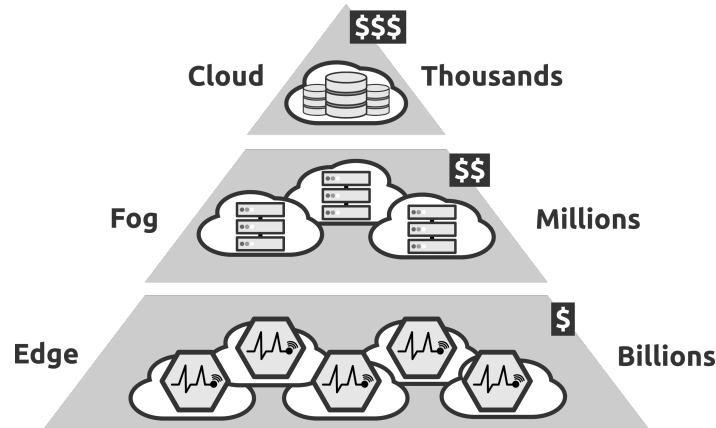
Figure 2.1: IoT, Fog and Cloud.

user, IoT is expected to directly impact him. In fact, IoT aims at sending or receiving data from objects by mean of sensors and making them available on the Internet. We can expect billions of devices connected to the Internet by the end of 2022 [6]. Thus, IoT platforms fall in the category of the near user platforms. Since they are in charge of sensing and interacting with the environment they are part of what is called the perception layer of Internet [12]. The technology used by the connected objects are highly dependent on the use case and, IoT platforms are thus considered as one of the most heterogeneous network platform. A perfect illustration of such heterogeneity is the FIT IoT-Lab testbed [3] which comprise thousand of IoT devices deploy on 6 different sites in France for research purpose. Despite this heterogeneity, IoT devices could be categorize as either a sensors or an actuator. First, sensor nodes are used as data sources to monitor physical phenomenons. The generated data will usually go to an aggregator node in charge of offloading the data and makes it available on the Internet. Second, the actuators are another type of connected object with the capability of being controlled by incoming network requests. Since data can now be generated by objects, IoT supply a new form of network communication called Machine-to-Machine (M2M). While most of the network communications where driven by humans, 31% of future communications will expect to be between machines [4] by 2022. To handle this emerging traffic, a new paradigm called Fog Computing has been created. Its goal is to gather data from the connected objects closer to the end user by deploying geo-distributed devices at the edge of the network [13]. To this end, data can be processed and filtered from the edge of the network and thus reducing the overall Internet traffic and delay for low latency applications. Similarly to IoT, Fog Computing platforms are near the user and are characterized by heterogeneous devices. Thus IoT and Fog

computing introduce many heterogenous nodes to the existing network. This further reinforces the difficulties of experimental studies.

2.1.2 Large-Scale Platform Based Applications

This section presents several non exhaustive use cases that can be impacted by the research on large-scale network platforms along with the difficulties that they raise.

With the explosion of streaming services more than half of the overall downstream traffic is related to video streaming [14]. As an example, Netflix is a popular video streaming provider. A single Netflix server is able to send around 30TB of data at 3Gbps (in average) per day [15]. To answer to this network resource crisis, video steaming providers heavily rely on data centers and more specially on CDN. It consists in several computers spread over the world and connected together to distribute specific content closer to the end user. This allow to reduce latency and improve the bandwidth. CDN are large-scale platforms and cover very large geographical area such as the Akamai CDN [16, 17] which comprise around 300 thousands servers spread over 136 countries. Netflix relies on CDN to place and replicate its video content. Its catalogs contains petabytes of data. It is therefore impossible to synchronize the complete catalog over the entire world, for network performance and data storage reasons. To this end, during off-peak hours, Netflix strives to predict which title is more likely to be used on a given region. In this way, only sub-parts of the catalog is present at a given location and network traffic is reduced. But other strategies are also implemented by Netflix to limit its impact on the network. Indeed, Netflix CDN called Open Connect, provides its own type of servers located at strategical points over the world as depicted on Figure 2.2. These servers can be present inside ISP to reduce the load but also between ISP on points called Internet eXchange Point (IXP). Other data placement strategies are also considered in research, such as moving data cache to the edge of the network can be envisioned [18]. This modern use case shows that current network usage implies high bandwidth applications with many nodes connected to a common network.

This tendency to make the end user more and more connected along with its environment is the essence of Smart Building and Smart Home. In current Smart Home, smart speakers are common connected objects. As a matter of fact, Google claims that it is selling 1 Smart Speaker every seconds [20]. Smart Speakers are voice-based virtual assistants which allow the user to retrieve informations or manipulating actuators [21]. Their network platform are composed of three components: 1) The Smart Speaker 2) The actuators (IoT devices) 3) The Cloud. The Smart Speaker is in charge of collecting the vocal commands and it is connected to the Internet usually via 802.11 (Wi-Fi). Then, the commands are sent to the Cloud and processed by the
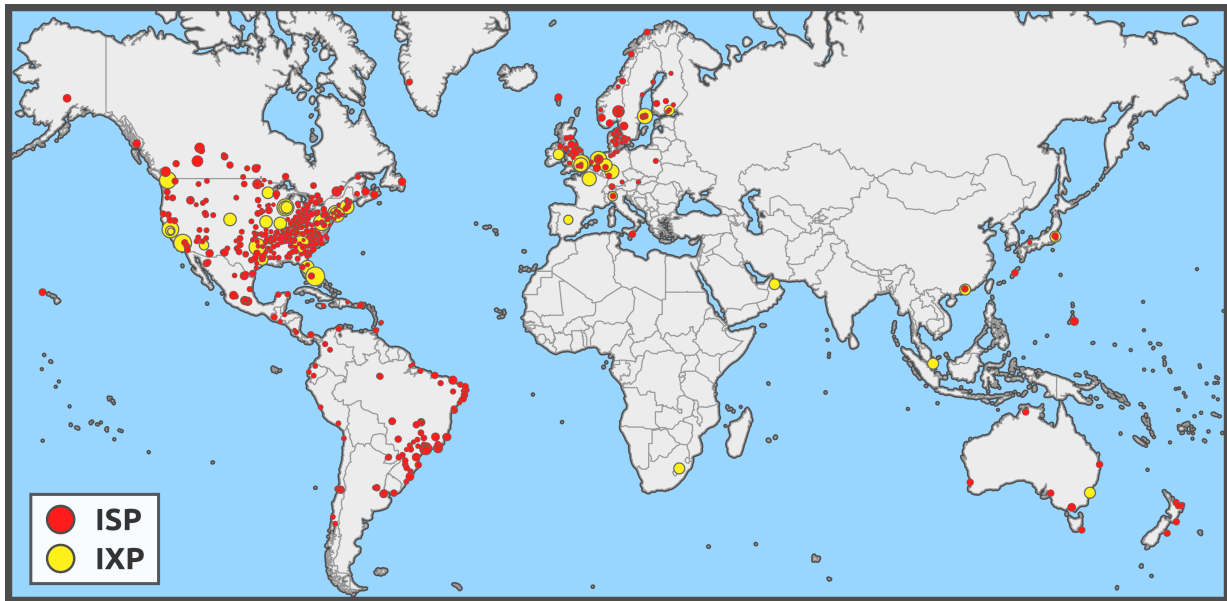
Figure 2.2: Netflix Open Connect deployment inspired by a recent publication [19]. ISP are Internet Service Provider deployment whereas IXP are Internet eXchange Point.

service provider' servers. Thanks to this architecture, the Smart Speaker can benefits from many services available on the Internet. Most of connected objects rely on this architecture and numerous of them are available on the market. Smart Bulbs [22] actuators which allow the user to switch on and off the light from anywhere in the world. In the same domain Smart Plugs allow to monitors and control the energy consumption of home devices. This constitutes a very promising solution to energy efficiency [23] and research in this domain proposed interesting feature: Device Identification, Voltage Control, Thermal and Overload Protection, etc. Other energy saving solutions such as Google Nest Thermostat [24] allow to regulate the home temperature using several sensors which comprises temperature sensors and ambient light or near field activity sensors. Since it is meant to be autonomous, it sends data periodically to a distant server in contrary to Smart Speaker. Thus, the network footprint for a given connected object seems to be highly dependent of its use case [25, 26] and varying from sporadic to high bandwidth streaming communications. These use cases demonstrate that a common usage of IoT involve different platforms such as Cloud or ISP network. Thus, it is difficult to study them separately since their are closely interacting together.

In addition, the IoT paradigm is also the building block of the concept of Smart Cities where this ecosystem of connected objects allow for improving citizen life comfort. As an example, Smart Grid [27] has promising impact on reducing the energy consumption. In fact, Smart Grid

is adding a network layer above the actual electrical grid. It allows for fine grain energy consumption monitoring. Consequently, it improves the forecasting abilities of energy providers. Moreover, this communication layer makes the grid more resilient by improving issues detection and prevention. In addition to Smart Grid, Wireless Sensors Networks (WSNs) [28] will play an important role in Smart Cities at sensing the environment. It consists in deploying numerous of nodes working in harmony with the aim of sensing physical phenomenons. Hence, the analysis power resulting from the combination of hundred of devices make them suitable for many applications involving large field coverage. For instance, air pollution monitoring [29, 30] based on WSN allows wide-range air quality monitoring. Sensors placed on the nodes are in charge of collecting the air quality data and sending them to a database. Thanks to the diversity of the measurements and their location, the air quality can be analyzed more precisely. Moreover, WSN are not limited to Environmental Monitoring. Thanks to there versatility, WSN can be applied to various domains [31] such as: military (battlefield surveillance, combat monitoring, intruder detection), health care (motion sensors, position sensors), urban (transportation, acoustic monitoring), industrial (corrosion, heat, vibration and position sensors). WSN contains numerous nodes which are targeting energy efficiency. Thus, WSN mainly rely on low power wireless communications. Consequently, we expect that the deployment of Smart Cities such as the SmartSantander testbed [32] will lead to a new generation of dense platforms composed heterogeneous nodes based on various communication technologies.

Large-scale platforms are involved in many domain of applications in which we can extract 3 critical properties. First, they can involve high bandwidth application. Second, they can be closely linked since they can communicating together from end-to-end. Finally, these platforms can be very large and dense. This reinforce the difficulties to study these networks and raise new research challenges which will be addressed in the next section.

2.1.3 Challenges

In the last section, we have seen that large-scale platforms offer great solutions for a variety of applications. Nonetheless, they raise a lot of challenges in many different area of research: security, wireless technology, network protocols, energy consumption, etc. In this section, we will skim over some of these areas to provide a general view of the research challenges raised by these platforms.

In ICT, security is based on three principles namely Confidentiality, Integrity and Availability also known as the CIA triad. As details previously, connecting hundred of nodes close to the end user has several benefits in terms of delivered services. Still, such a node deployment has an

impact on the security and affects the following layers of the IoT architecture [33]: 1) Perception layer (nodes themselves) 2) Transportation layer (network between the nodes and the data recipients) 3) Application layer (customers service provider). First, the perception layer is the most vulnerable point since the nodes and the network it belongs is potentially accessible by the attacker. Indeed, nodes can be damaged, malicious code can be injected and data can be stolen or corrupted. In addition, current IoT networks suffer from identification problems. Thus, they can be targeted by spoofing attacks (impersonation) where malicious nodes behave such as a normal node as part of the network. Next, transportation layer is vulnerable to integrity and confidentiality attacks and also to more esoteric attacks such as routing attacks where malicious nodes are able to modify the routing path of the data being transferred. Finally, the application layer can be subject classical ICT attacks such as Denial Of Service (DoS) or data leakage.

But IoT security problems are also driven by the lack of standardization. As a matter of fact, the development of IoT cover a very large research area involving tremendous technologies as depicted on Figure 2.3. On the network layer of the TCP/IP model, nodes can communicate in various way involving different type of technologies: Low Power Wide Area Network (LPWAN), Wireless Body Area Network (WBAN), Low Rate Wireless Personal Area Network (LR-WPAN), etc. In addition, the Internet layer uses addressing and routing mechanisms whereas the application layer includes complex end-to-end protocols. This diversity prevents interoperability between IoT platforms [34, 35]. Indeed, since there is no standard way to identify and communicate with different IoT platforms, we cannot rely on profiling to make these platforms available on the application side. Consequently, applications are often tied to a specific type of IoT platform.

Connecting numerous of nodes together imply to deal with scalability issues [37]. Starting from the deployment, nodes should be easy to setup and configuring each node individually is not an option. Thus, automated bootstrapping solutions which can involve network communications [38] should be provided to configure the IoT platform automatically. On the network level, communication technologies used in the platform should be able to handle massive concurrent medium access while maintaining good performance. With this aims, Wi-Fi 802.11ax was developed to achieve better performance in dense scenario [39] especially with the adoption of Orthogonal Frequency-Division Multiple Access (OFDMA). Apart from the edge side, the large amount of data generated by these new devices also impact Cloud environment. Current databases technologies has to deal with query on massive data [40] and servers should maintain sufficient computational performance to ensure good Quality of Service (QoS) [41]. Consequently, all these scalability issues should be addressed by researchers. However, con-
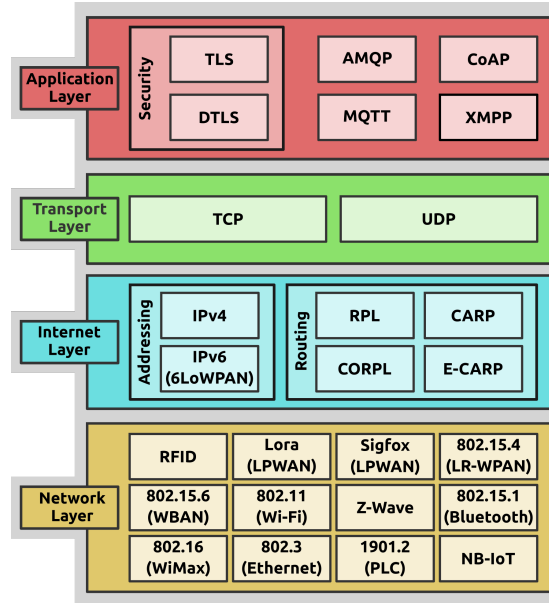
Figure 2.3: Non-exhaustive representation of network IoT communication technologies using the TCP/IP model. This Figure is inspired from [36].

ducting such experimentations requires to deploy numerous node which is expensive and time consumming. Similarly, as detailed in Section 2.1.2, a common network architecture involves connecting IoT devices to the Cloud which leads to even more complexity.

Another big challenge encountered on large-scale platforms is the energy consumption. As an example, the energy consumed by data centers in the United States represent 1.8% of the total amount of energy consumed by the country [43]. Since data centers contain thousands of ICT devices they require dedicated cooling systems which generate additional energy cost. Thus, as depicted on Figure 2.4 the energy consumed by the ICT devices represents around 50% of the total data center energy consumption. Hence, reducing the energy consumed by the cooling systems is espected to provides more energy efficient data centers. To measure this efficiency, a metric called the Power Usage Effective (PUE) has been introduced [44]:

$$\text{PUE} = \frac{E_{\text{DataCenter}}}{E_{\text{ICT}}} \qquad \text{with PUE} \in [1, +\infty[$$

Besides cooling systems, servers are the next energy-hungry components. To improve their energy efficiency, multiple solutions as been proposed [45]. First, servers can use better computer architecture. In fact, it has been demonstrated that the Reduced Instruction Set Computer (RISC) architecture consumes 3 to 4 times less energy compare to the common Complex In-
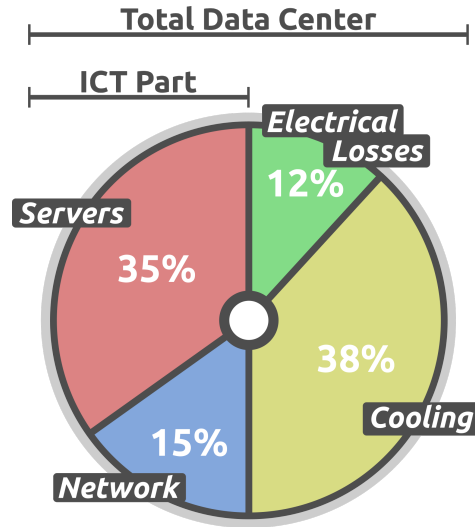
Figure 2.4: Data center energy consumption repartition according to an air conditioning study [42].

struction Set Computer (CISC) architecture [46]. Next, techniques based on Dynamic Voltage and Frequency Scaling (DVFS) allows to regulate CPU frequency to better control their energy consumption [47, 48, 49]. Other works introduce Power Capping which allows to define power consumption limits on servers [50]. Finally, research on turning off servers are also conducted [51]. Thus, data center energy saving cover several research domain and conducting real experiments require access to dedicated testbeds.

Regarding IoT platforms for power constraint applications, hardware energy consumption is often related to communication [52]. Indeed, various phenomenons can modify the communications' energy consumption [53] such as: Radio transmission and reception, collisions, overhearing (receiving packets while being the wrong recipient), control packet overhead, idle listening, over-emitting. Thus, many wireless technologies is dealing with the trade-off energy/performance and strive to offer the best of both optimization axes. Energy harvesting [54] is also a good solution to extend battery life and implement autonomous nodes. Energy can be extracted from various (and sometimes surprising) sources: solar energy, radio frequencies, wind, motions, temperature or even breathing. Nonetheless, focusing only on hardware energy consumption is not the right way to go. In fact, up to 80% of the hardware energy consumption can be driven by the software [2]. But achieving software energy efficiency is really challenging since various layer of abstraction has been added between the software and the hardware to simplify softwares development [55]. For example, software energy profiling solutions should be

able to deal with multi-threading which is highly tied to hardware. Additionally, source code compilers may generate different binary depending on its version. Thus, the same application can have a totally different impact on performance and energy consumption according to the compiler in use.

Still, different methods exist to obtains software energy footprint. For example, code instrumentation allows to extract execution traces at runtime and hardware Performance Measuring Counters (PMC) [2] allows for performance measurements. Other measurement techniques exist such as wattmeters [56] but it relies on dedicated platform. Moreover, all these solutions are difficult to employ for large-scale platform studies. It requires to monitor multiple nodes at runtime and implies hundred of network communications.

Today's and future large-scale platforms introduce a lot of challenges in the research domain. The energy consumption account for major part and are tied to all the large-scale platforms. But conducting real experiments becomes more and more difficult since it involves many nodes and technologies. Additionally, these large-scale platforms are often interracting with each other from end-to-end. This call for a convenient experimentation solution to study large-scale networks energy consumption from end-to-end.

2.2 Simulation Models

Modern network platforms are very large and consume a lot of energy. They raise many research challenges and are difficult to study by mean of real experimentations. As stated in the Introduction, network simulation is a great alternative to real experimentation. They allow for time, money and energy saving specifically on large-scale platforms. These large-scale platforms are often interacting together to form the end-to-end network which comprise the IoT, the Fog, ISP and the Cloud. Hence, studying them together is very convenient from the research point of view but it is even more challenging.

Thus, our goal is to propose a scalable simulation solution to study the end-to-end network energy consumption with the aims of reducing its environmental impact. This goal implies to be able to predict the time and the energy consumption of three physical tasks namely the CPU computations, the wired communications and the wireless communications. Both time and energy models are closely related. Energy models are based on their counterpart time prediction models (or performance models). This close relation between energy and time

prediction is explained by the following formula:

$$E(T) = \int_0^T P(t)dt \tag{2.1}$$

In this section, we focus on the state of the art of the models which are involved in the contributions of this thesis: the wired performance model, the wired energy model and the Wi-Fi performance model. This stack of models are summarized on Figure 1.2. We now detail how these models work in network simulators with the aim of providing the required background.

2.2.1   Wired Performance Models

The primary goal of a network simulator is to predict network communication durations with potentially multiple computers. At the beginning of the Internet, most of the communications where wired and current platforms which require high bandwidth and low latencies connections are still using wired connection such as data centers and ISP networks. Consequently, the majority of network simulators usually propose a wired performance model.

A network link can be modeled by two properties. The first one is called latency. It represents the delay from end-to-end on the wire. The second metric is the bandwidth. It represents the amount of information that the link is able to carry for a given duration. Considering a single network communication between two hosts to transmit N bytes of data on a wire with a latency L (in seconds) and a bandwidth BW (in bit per seconds). We can deduce the communication duration T with the following formula:

$$T = L + \frac{N \times 8}{BW} \tag{2.2}$$

Thus, knowing the link characteristics, the next step is to deduce the data size N. This data size depends on the protocol stack used by the transmitting machine. Considering a simple Transmission Control Protocol (TCP) socket communication, one possible protocol stack is proposed on Figure 2.5. Note that, this network stack was chosen because it is massively used by today's wired platforms and it is commonly present in networks simulators. The data link and physical layer are implementing the Ethernet features which allow for communications over the wire. These features include the Ethernet protocol and the physical signal properties. The network layer implements the IPv4 protocol in charge of identifying the machine on the network. Finally, the transport layer implements the TCP protocol in charge of connection management and flow control. Thus, knowing this protocol stack and the payload P that we
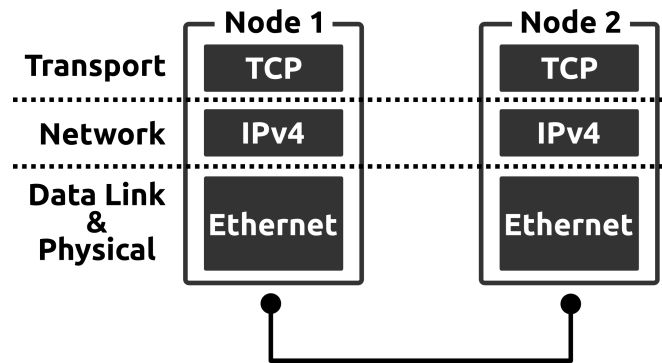
Figure 2.5: Simple TCP connection based on the 4 last layers of the OSI Model.

want to transmit, network simulators are able to compute the effective amount of transmitted bytes over the wire. Since the payload P pass through each layer of the stack before being sent over the wire, N can be computed by summing P with the overhead induced by each layer of the stack.

Once we can estimate communication's duration, the next step is to provide bandwidth sharing mechanism. This allows to simulate scenarios with multiple machines that communicate over the same wire. On real networks, this task is handled by the current TCP protocol implementation and the different network devices between the sender and the receiver [57]. Regarding TCP, the protocol strives to provide fairness among the communications. Its implementation contains a congestion control mechanism in charge of using the right amount of bandwidth according to the network conditions. This algorithm comes with several flavors such as: Tahoe, Reno, New Reno, Westwood, Vega, BIC or CUBIC (the most common one on recent Linux operating systems) and many others. Consequently, the fairness index [58] of each implementation is different. To solve this issue, network simulators can choose to implement a full TCP version (or several) to model its sharing impact over the network. Nonetheless, some network simulators have the strong assumption that on a given link, TCP is perfectly fair [59, 60]. This assumption provides multiple benefits such as a reduced complexity and performance improvements. Still, each network protocol converges toward a specific fairness equilibrium. Modeling other protocols such as User Datagram Protocol (UDP) thus requires a careful study of their behavior [57].

2.2.2 Wired Network Energy Model

This section introduces wired network energy model based on the time predictions presented in previous section. Indeed, communicating through a wire require energy which depends on several factors. Let's first introduce how ICT devices are consuming energy.

Generally, an ICT device (CPU, router, switch, etc.) is not subject to a constant load. As an example, CPU usage may vary over time. This variation of load has an impact on the power consumption. Thus, an ideal ICT device should not consume any nergy when no load is applied to it, and should consume its peak power when working at maximum performance. This would be the ideal model. Nevertheless, in reality any ICT device consumes energy even when it is idle. Thus the energy consumption of ICT devices can be decomposed in two parts:

- **Static:** Power consumed by the device when it is idle (noted P_{static})

- **Dynamic:** Additional power consumed while the device is under load (noted P_{dyn})

Depending on the considered device, the static part of the energy consumption might turns out to be significant. As an example, a server can consume more than half of its full power when idleing [61]. On the other hand, the dynamic energy consumption is often considered to be proportional to the load of the device (noted S_{load}) in many models, and thus $P_{dyn} \propto S_{load}$. But this is not true on real devices. For example, CPU power levels are discrete and thus, a better approximation P_{dyn} for such a system must be discrete. More formally, considering a system with n power levels. Let's define $p_1, ..., p_n$ the dynamic power generated by each power level associated with their respective load intervals $\{I_1, ..., I_n\}$. In reality, the total power of the system P_{total} is express as follow:

$$P_{total}(S_{load}) = P_{static} + \underbrace{\sum_{k=1}^{n} p_k \times \overbrace{\mathbb{1}_{I_k}(S_{load})}^{\text{Indicator Function}}}_{\text{Dynamic Power}}$$

The ideal, proportional and real model are represented on Figure 2.6.

A similar approach can be followed regarding network devices. First, P_{idle} corresponds to the power consumed by the network device while no communications occur on the wire. However, P_{dyn} varies from device to device depending on the number of port and their respective performance. Hence, we usually employ the term "port power consumption" in reference to the dynamic power of a network device. The dynamic energy consumption of a network device can be decomposed in several parts:
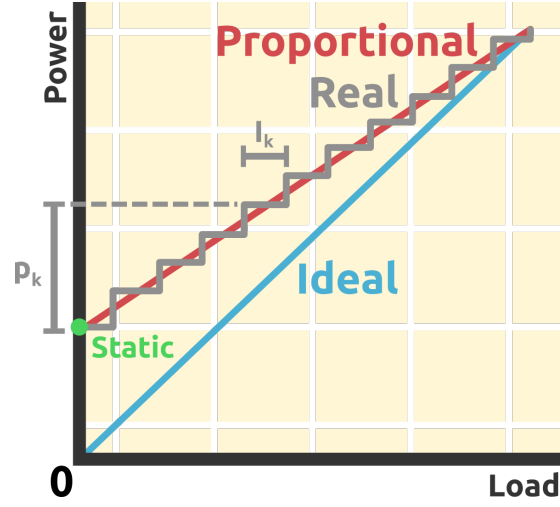
Figure 2.6: Ideal, real and proportional model of ICT devices power consumption.

- E^{ptx}: Transmit packet energy consumption

- E^{prx}: Receive packet energy consumption

- E^{btx}: Transmit byte energy consumption

- E^{brx}: Receive byte energy consumption

Thus, considering a network device with a packet processing rate R^p and n ports, each of them working at an effective byte processing rate of R_k^b. The entire device power consumption can be written as follows:

$$P_{netdev} = P_{static} + \overbrace{R^p(E^{prx} + E^{ptx})}^{packet} + \underbrace{\overbrace{\sum_{k=1}^{n} + R_k^b(E_k^{brx} + E_k^{btx})}^{ports}}_{dynamic} \qquad (2.3)$$

Note that each port has its own byte energy consumption. This is due to the fact that each port can have different characteristics (100Mbps, 1Gbps, 10Gbps, etc.) and consequently different energy consumption values. These energy values are difficult to measure. It requires specific hardware and high precision measurements tools [62, 63]. That is why, using energy measurements from the literature along with network simulations provides and efficient solution to study the energy consumption of network devices.

2.2.3 Wi-Fi Performance Model

Wi-Fi is a common wireless technology used mainly at the edge of the network. Simulating Wi-Fi communications at fine-grain requires to account for the Wi-Fi Medium Access Control (MAC) layer, the physical layer and the communication channel. In this section, we present the Wi-Fi MAC layer based on 802.11n standard [64] since its features are implemented by most simulators and available on today's implementations. It is followed by a brief introduction to communication channel models used in common network simulators.

Wi-Fi MAC Layer

The Wi-Fi MAC layer has two operating modes: Ad-Hoc and Infrastructure. Figure 2.7 presents these two operating modes. In this thesis, we focus essentially on the Infrastructure mode, that happens to be the most commonly used mode. In this mode, Stations (STAs) are connected to an Access Point (AP). The STAs use the AP as a gateway to the Internet and to the other stations.

According to the IEEE 802.11 Specification [64], the MAC layer of the 802.11 standard is using a Distributed Coordination Function (DCF) based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to access to the medium. In the context of Wi-Fi in Infrastructure mode, this medium access mechanism provides a way for the stations to initiate a communication to the access point. To transmit a packet, a given station first has to sense the channel before transmitting and for a given period of time called Distributed InterFrame Space (DIFS). If the channel is sensed idle during a time DIFS (Carrier Sense), then the station can send its packet to the access point. However, if the channel is sensed busy (Collision Avoidance) then the station wait for a random time called backoff time before attempting for
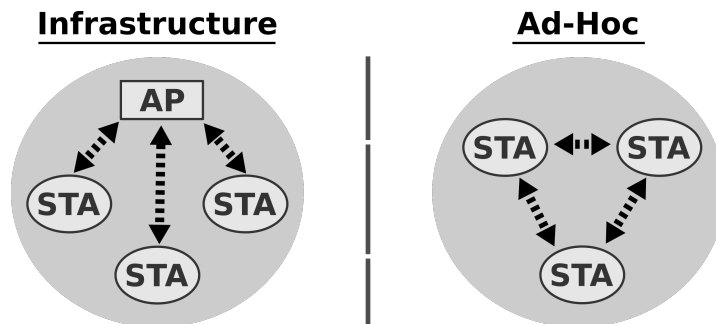
Figure 2.7: Wi-Fi 802.11 operating modes. Stations (STA) and Access Point (AP).
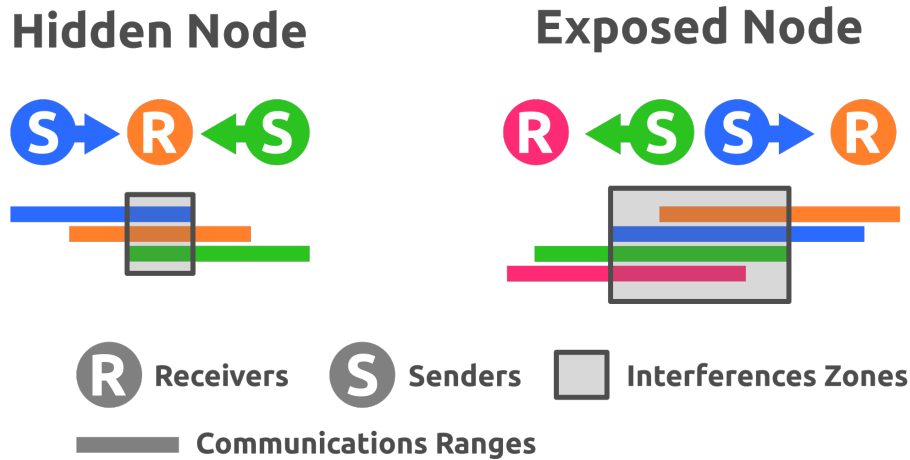
Figure 2.8: Basic CSMA/CA medium access problems.

another transmission. The backoff time is increasing exponentially according to the number of transmission failure and decrease in a similar fashion. Next, to ensure that a successful packet transmission, the access point sends an acknowledgment (ACK) to the station after a Short InterFrame Space (SIFS). It is important to note than $T_{DIFS} > T_{SIFS} + T_{\text{ACK Propagation Delay}}$. In this way, data can be transmitted during a DIFS without impacting the performance since the stations are able to sense the channel idle even during the transmission of the ACK. Then, to ensure fairness among the stations, after a successful transmission a station has to wait for a random backoff time to avoid channel capture.

Besides basic wireless medium access, CSMA/CA reveals some drawbacks when it comes to interferences. The first problem is known as the hidden node problem, and the second as the exposed node problem (Figure 2.8). The former arises when two STAs are trying to communicate with the same AP while being mutually out of range. Thus, each station cannot sense the channel idle. Consequently, they start their transmission and their respective signal interfere on the AP resulting in communication failures. The latter problem occurs when two stations in their respective cell are interfering with each other. Thus, each of them could sense the channel busy even if their respective AP could perfectly receive the signal. To solve these issues, an optional handshaking mechanism based on Request To Send (RTS) and Clear To Send (RTS) messages can be trigger by the DCF. Thus, if a station do not received CTS response from its destination, it can deduce that the receiver can face to interferences (hidden node) and delay its transmission. On the other hand, if a station receive a RTS from its neighboring nodes and do not receive their corresponding CTS, it can infer that it is an exposed node. In this way, RTS/CTS mechanism help to achieve higher throughput, particularly on dense scenarios.
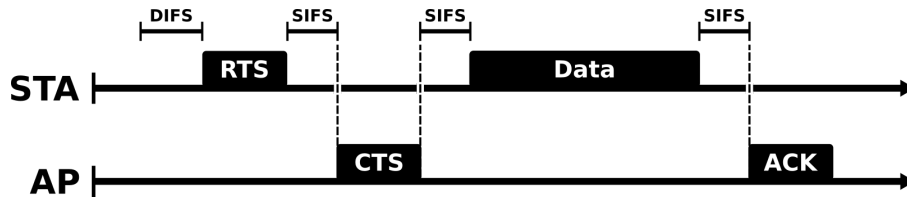
Figure 2.9: Wi-Fi communication diagram between a station (STA) and an access point (AP) using the RTS/CTS handshaking mechanism.

As an example, a communication between a STA and an AP based on RTS/CTS is depicted on Figure 2.9.

Wi-Fi Physical Layer

Simulating Wi-Fi involves numerous physical parameters. Regardless of the communication channel profile, Wi-Fi communication performance are determined by 5 parameters: 1) Modulation 2) Bandwidth 3) Coding Rate (CR) 4) Number of Spacial Stream 5) Guard Interval size (GI) (NSS). A specific combination of the first 4 parameters is known as a MCS. Each MCS has its own index which can be used to refer to a specific physical configuration. Lets first introduce the meaning of these five parameters.

1) A single Wi-Fi data sub-carrier is using phase and amplitude modulation to carry the data. Modulations range from pure phase modulations such as Binary Phase-Shift Keying (BPSK) or Quadrature Phase-Shift Keying (QPSK) up to 64 Quadrature Amplitude Modulation (QAM). Each of these modulation produces a constellation diagram as shown on Figure 2.10. This figure shows that higher granularity modulation will improve performance (higher symbol rate) but at the same time reduce the EVM box size. Thus, the signal will get more subject to error on poor channel conditions.

2) The bandwidth of a signal determines the frequency spectrum range of that signal. Wi-Fi is using Orthogonal Frequency-Division Multiplexing (OFDM) which makes the performance very sensitive to the bandwidth. Indeed, OFDM allows to send multiple signals in parallel. This is possible as long as the signals are orthogonal to each other which allows to separate them on the receiver side. To this end, doubling the Wi-Fi bandwidth allows to use two times more data sub-carrier and thus double the throughput. The 802.11n standard is using 20MHz channels and comes with a feature called channel bonding which allows to combine two 20MHz channels and reach the performance of a 40MHz channel.

3) The coding rate account for the proportion of useful data that is actually transferred

(a) BPSK.   (b) QPSK.   (c) 16-QAM.
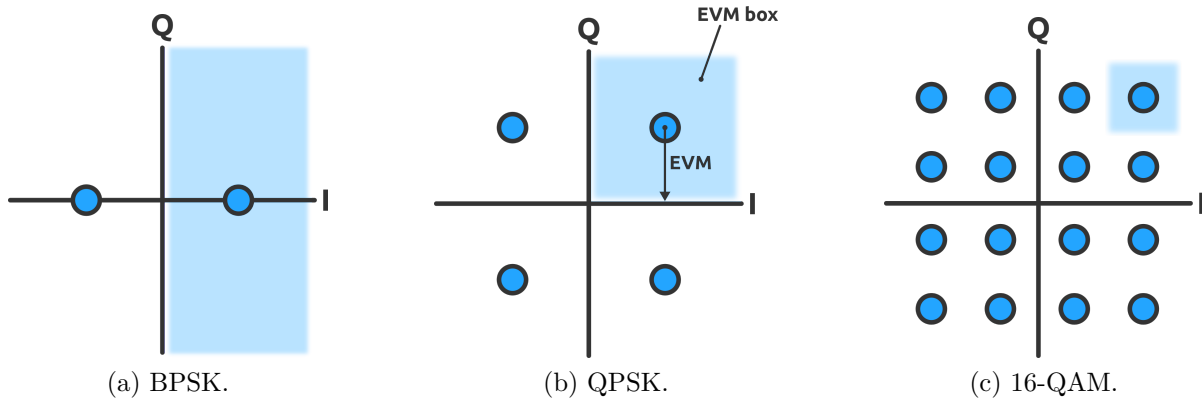
Figure 2.10: Constellation diagram of different Wi-Fi modulations with their Error Vector Magnitude (EVM) box.

during a communication. As an example, a communication with a $CR = \frac{7}{8}$ sends 8 bits to transfer 7 useful bits of data. This additional data sent, add error correction bits to the data. It improves thus the communication resilience and ensure data correctness.

4) Most of Wi-Fi standards offer Multiple Input Multiple Output (MIMO) communications. This feature allows to send multiple signal at the same time on different antennas. This technique called spacial multiplexing exploits the receiver signal diversity (several antennas) to separate the interfering signals. Thus, a Wi-Fi communication with a NSS of 2 is going to be twice as efficient as a one with a NSS of 1 which improves the spectral efficiency.

5) Adding Guard Intervals to a signal reduces its probability to be corrupted by another one. In the case of OFDM, a GI is added before each symbol that is transmitted. To this end, as long as an interfering signal fall into this interval, the data will still be unaffected by the interferences. Thus, increasing the GI improves the signal resilience but reduce the overall throughput (lower spectral efficiency).

Wi-Fi Communication Channel

During Wi-Fi communications, the signal get affected by the wireless channel which can lead to a significant variation of performance. Thus, communication channel models allow to account for these variations. The communication channel models are composed of a propagation delay model, a propagation loss model, an error rate model and an interference model. The propagation delay model is usually based on the light speed in vacuum $c = 2.99e8ms^{-1}$. The propagation loss model account for the signal power loss during its propagation. It can be based on the Friiz propagation loss model for line of sight communications or log distance model for

indoor scenarios. The error rate model takes into account the modulation resilience to the thermal (such as the Johnson-Nyquist model [65]) and background noise. Finally, the interference model account for wireless communications that occur concurrently. They are based on a simple additive power model [66].

Estimating the end-to-end network energy consumption requires to combine multiple models which comprise performance and energy models. Performance models predict the tasks' durations such as wired and wireless communications. These models are based on low-level properties of the network such as packets, medium access protocols and physical channel properties. On the other hand, energy models predict the energy consumed by these tasks. They are based on their counterpart performance model. Estimating the energy consumed by wired communications involved low-level properties such as packet and byte energy consumption. Since the performance and the energy models of common network simulator are fine-grain, they fail to scale on large-scale platforms and high bandwidth application scenarios.

2.3 Experimentations and tools

Scientific experiments are vital to understand our world. They allow to build models for complex systems and potentially predict their evolutions. Models can be built inductively. This imply to study a particular case by mean of experiments and then constructing a model by generalization. On the other hand, models can be constructed by deduction. In this case, several hypothesis are proposed for a system. Then, these hypothesis can be accepted or rejected by mean of experiments. Thus, experimentation is essential in the process of research since it is the cornerstone of the models.

Regarding Computer Science, experimentations can be conduct by following different approaches. Each of them has advantages and drawbacks as we will see in this section. Since our work is based on network simulations this section also tackle the two types of network simulator namely Packet-Level Simulator (PLS) and Flow-Level Simulator (FLS).

2.3.1 Computer Science Experiments

In Computer Science research, there are many ways to conduct experiments. They can be categorized into three types: in vivo, in vitro and in silico experiments. First, in vivo experiments consists in studying a system by mean of observations. This allows to avoid the alteration of the system by the observer and it is considered has the most trustful type of experiment. Then,

in vitro experiments strive to reproduce the system in a controlled environment such as virtual machines. Consequently, experiments can be reproduce (not necessarily identically). Thus, it offers more flexibility than in vivo experiments. Finally, in silico experiments is a neologism employed to designate experiments that occur on computers. This type experiment can be used to study many system from various field of research. It offers even more flexibility than in vitro experiments, but requires to construct solid models. The rest of the section (and the entire manuscript) focus on in silico experiments.

In silico, or computers experiments can be categorized as ether Emulations, Simulations Hybrid or Hardware In the Loop (HIL). Emulation consists in reproducing a complete system on a computer to study it. This category, mainly applied to Computer Science systems since natural phenomenon cannot be perfectly modeled. On the other hand, simulations consist in constructing models which are accurate enough to fully represent the original system. For example, many physical systems such as the n-body problem can be simulated offering sufficient environment to conduct studies[67]. Next, hybrid experiments strive to use the best of both categories (emulation and simulation) on a single experiment [68]. Finally, HIL experiment is at the frontier between in vivo and in silico experiments. It consists in making hardware devices interacting with the simulation environment. In this way, systems which are complex to model could be added to the experiment loop while saving time and having better accuracy [69, 70, 71].

In this thesis, we are focusing on in silico simulations meaning that our work do not involve hybrid or HIL experiments. In fact, since were targeting large-scale platform simulations, restricting our work to simulation allows to benefit from the scalability property.

In addition to scalability, we are interested in experiment reproducibility. In Computer Science, this feature allows to reproduce scientific works by making all the experiment process publicly available. Reproducibility increase the reader confidence in a work and can even help him/her in their own research. However, many work which claim to be reproducible are in reality hardly reproducible because of many factors [72] such as author unavailability, artifacts unavailability, unclear documentation or are only partially reproducible. But reproducibility is not a trivial task to achieve due to several factors [73]. First, reproducibility cannot be achieve if it is not possible to reproduce the exact same binary file(s) which has generated the results. Since common executable has a wide dependency tree, this problem is be difficult to address. Several package managers such as Guix [74] or Nix [75], aims at solving this issue. They can ensure to provide the exact same dependency tree used during for the experiment. In this way, we ensure that compilers could generate the exact same binary files. Still, learning these tools

is time consuming and can restrain authors [73].

In this the thesis, we claim our models to be scalable and accurate and hence these two properties should be verifiable. Thus, we strive to provide reproducible experiments with available artifacts along with all the steps to reproduce them. Our experiments are using simulators of networks and distributed applications as a core. The remaining of this section describe and compare such simulators.

2.3.2   Application-Level Simulators

Application-Level Simulator (ALS) is a type of network simulator oriented towards the simulation of distributed applications. ALSs propose convenient interfaces to represent distributed applications. They allow to study the interactions between the different distributed components of the application. Although prediction accuracy is often presented as an important ideal goal, the tools presented in this section put a greater emphasis on the modeling of the application than on the accuracy of the network performance prediction. Most of them rely on simple latencies/bandwidth models as presented in Equation 2.2. Network protocol (TPC, UDP, etc.) overhead are thus not taken into account. In this section we detail several ALSs to evaluate their applicability for end-to-end network energy consumption studies.

CloudSim DES written in Java [76]. It provides a Java framework to setup and simulate Cloud networks. Since it provides packets as transmission unit. Yet, it does not provide any protocols stack implementation such as TCP or UDP. CloudSim proposes federated Clouds model which consists in orchestrating interconnected data centers to share the resources for the end users. CloudSim is representing the applicative part by means of a Cloudlet interface which encapsulate the application resource profile. Since CloudSim target essentially cloud simulations it used for different purpose such as economical study, provisionning policies, service delivering policies and energy consumption. Regarding energy consumption, CloudSim is limited to servers and VMs studies. Moreover, it suffers of network bandwidth aberrations as stated in [77]. Consequently, to overcome the main issues of CloudSim, DartCSim+ where proposed [78]. DartCSim+ is an extension of CloudSim. It allows for wired network energy estimations and improves the VMs migration network model. Still, CloudSim and DartCSim+ cannot be used outside of the Cloud context to simulate various types of network topologies. To this end, they could not be used as an end-to-end energy consumption framework.

To simulate Fog environment, iFogSim has been proposed [79]. It is an extension of CloudSim which provides Fog applications simulation. New entities such as sensors, actuators and Fog devices has been added. Communication between sensors and Fog nodes occurs by means of

Table 2.1: Application-Level network simulators review.

Simulators	Domain	Performance			Energy		
		CPU	Wired	Wireless	CPU	Wired	Wireless
CloudSim	Cloud	✓	✓	✗	✓	✗	✗
DartCSim+	Cloud	✓	✓	✗	✓	✓	✗
iFogSim	FogCloud	✓	✓	✗	✓	✗	✗
mtCloudSim	Cloud	✗	✓	✗	✗	✗	✗

tuples with a simple bandwidth/latency model as presented in Section 2.2.1. Fog applications are represented by Directed Acyclic Graphs (DAG), where nodes of the graph represent an application module and each edge represent modules dependencies. In this way, applications module can be mapped to different nodes (either to the Cloud or to Fog depending placement policies) which allows for distributed Fog application simulations. Regarding energy consumption, iFogSim still benefits from servers and VMs energy models inherited from CloudSim and allows for Fog node energy measurement. Thus, iFogSim can be used to study coarse-grained Fog application placement strategies and their effects on performance and energy. However, iFogSim does not provides any wireless communication models and does not account for network protocols overhead. Despite providing an end-to-end framework, iFogSim is restricted to Fog application study and does not target accurate network energy predictions for neither wireless nor wired communications. Consequently, iFogSim is not suitable for end-to-end network energy studies.

mtCloudSim is a flow-level DES designed for multi-tenant Cloud simulations [80]. Multi-tenant Cloud consists in sharing the same application instance among different tenants (users) [81] to optimize resource sharing. Classical PLSs requires to mark each packets to identify each Cloud user while in mtCloudSim, user can be identify simply by marking flows. mtCloudSim proposes an energy model for network switches only and seems tied to Cloud studies. Thus it cannot be used for end-to-end energy consumption studies.

ALSs offer versatile distributed application network simulation. Taken separately, they cover a large part of today's network platforms. However, there is no ALS which covers the network from end-to-end with accurate network performance predictions. Indeed, ALSs network model suffer from a lack of validation. Consequently, existing ALSs cannot be used for end-to-end network energy consumption studies.

2.3.3   Packet-Level Simulators

Packet-Level Simulator (PLS) is a type of network simulators providing fine-grained network models [82]. They strive to model every aspect of common networks such as packets, protocols (TCP/IP,UDP, routing, etc.) and Physical layers (interferences, buffering, etc.). The Figure 2.11 shows a network platform as represented in PLS.
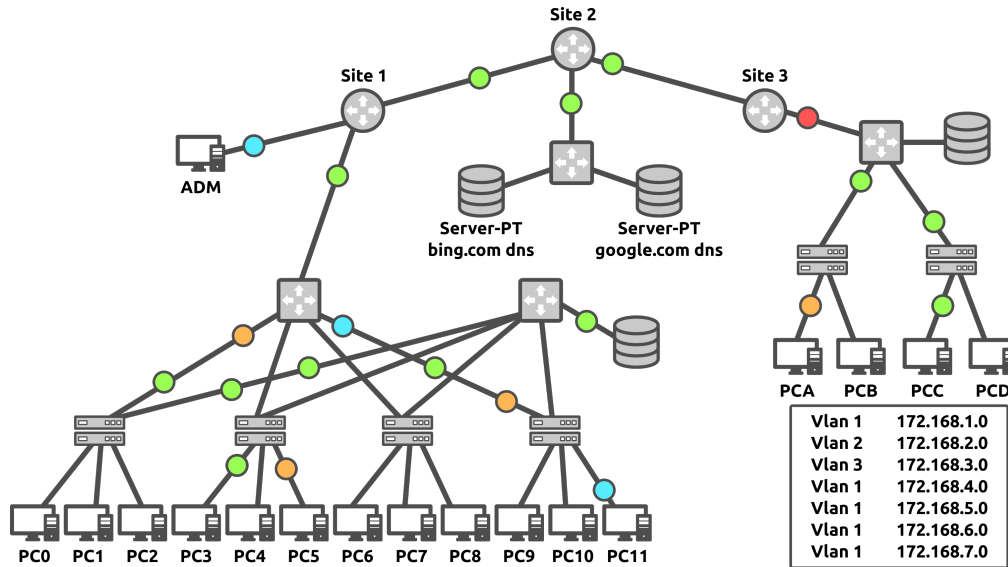
Figure 2.11: Example of a network platform as represented in typical PLS.

Thanks to their faithful representation of real networks, PLSs are often considered to be the reference in terms of predictions accuracy [83]. As presented in Section 2.2, although they provide accurate predictions, there are also harder to instantiate since they come with many parameters to setup [77]. Regarding performance (execution time and memory usage), PLSs suffer from scalability issues on large-scale platforms [84, 85]. In fact, as they name suggest, PLSs simulate every packet that flows through the network. This has two major consequences on the performance. First, increasing the number of nodes in the network can potentially leads a to larger number of communication and thus decreasing the simulation performance. Second, on scenarios involving high bandwidth applications, the number of generated packets increase drastically which leads to bad performance. Nevertheless, despite these drawbacks PLSs are still widely used in research [86, 87] for their fine-grained properties [88, 89, 90]. The remaining of this section is dedicated to the review of recent open source PLSs.

CupCarbon is a WSN network simulator [91] written in Java. It falls in the category of Discrete-Event Simulator (DES). DES are the most common network simulators where the

system state (simulated time, energy consumption, etc.) is changing according to a discrete event list resulting from the initial conditions [92]. CupCarbon is also agent based, meaning that each sensors are considered to be independent agents. These agents can be programmed in the CupCarbon Domain Specific Language (DSL) called SenScript [93]. Sensors can communicate through three communications technologies namely 802.15.4 (Zigbee, 6LoWPAN), Wi-Fi and LoRa. In addition, CupCarbon offers sensors mobility features and energy consumption predictions. CupCarbon targets WSN algorithms development, testing and debugging prior to real sensors deployment [94]. It is able to simulate hundred of nodes but still, is is tied to the WSN domain and it is not possible to use it outside of this context. Thus, CupCarbon is not suitable for end-to-end network energy consumption studies.

Komondor is a Wireless Local Area Network (WLAN) simulator written in C++ [85]. It is dedicated to simulate recent Wi-Fi standards. Currently, it implements the 802.11ax and support many of its features such as: 1) DCF 2) Aggregation 3) Dynamic Channel Bounding 4) MCS selection 5) RTS/CTS handshaking 6) Spacial Reuse. Komondor is dedicated to dense Wi-Fi communications study [95] and do not provide any energy consumption models. Consequently, it cannot be used for Cloud, ISP or IoT studies.

OMNET++ is a versatile discrete-event library written in C++ [96]. It is usually combined with the INET library [97] which proposes a complete network simulation environment. INET gives the different building blocks to simulate various network scenarios involving several wireless communication technologies such as Wi-Fi or 802.15.4. Additional frameworks could be added to integrate Bluetooth Low Energy (BLE) or LoRa communication models. Moreover, INET provide a fine-grained protocol stack implementations for various network protocol. Regarding energy consumption, INET provides the building blocks for nodes and radio components power measurements [98] but it has no wired energy model. Despite their flexibility, OMNET++ and INET are not suitable for large scale experiment such as the end-to-end energy consumption simulation. In fact, simulations are slow and could not met the requirement of today's large-scale platforms.

Finally, ns-3 is a DES written is C++ [99]. It proposes a complete C++ framework to simulate classical networks involving wired and Wi-Fi communications. Ns-3 implements a complete network protocol stack with a POSIX-like socket interface. In addition, ns-3 allows for power consumption measurement on node which take into account radio communications [100]. Wired energy consumption models are provided by an external ns-3 module called ECOFEN [101, 102]. This module allows for wired network energy predictions based on the static, packet and byte energy values. ECOFEN has been validated in the literature [103] which makes it suitable for

Table 2.2: Packet-level network simulators review.

Simulators	Domain	Models					
		Performance			Energy		
		CPU	Wired	Wireless	CPU	Wired	Wireless
CupCarbon	WSN	✗	✗	✓	✗	✗	✓
Komondor	Wi-Fi	✗	✗	✓	✗	✗	✗
OMNET++	Various[a]	✗	✓	✓	✗	✗	✓
ns-3	Various	✗	✓	✓	✗	✓	✓

a. Can be applied to various domains.

scientific studies. Thus, ns-3 provide great network simulation capabilities however, simulation performance are very bad on large-scale scenarios. Moreover, ns-3 does not provide CPU performance models. Thus, it cannot be used as an end-to-end energy consumption framework.

As presented in this section, many network simulators propose great features for network communication and energy simulations. However, none of them meet the requirements for end-to-end energy consumption study related to large-scale platforms. Table 2.2 summarizes this PLS analysis. In addition, a common weakness of PLSs is related to performance. In fact, all of them fail to scale considering a large-scale platform scenario with high bandwidth application. To this end, investigating another type of network simulator may gives us this scalability properties as we will see in the next section.

2.3.4   Flow-Level Simulators

To improve simulation performance beyond packet-level simulators presented in previous section, it is possible to follow three different approaches [83]. The first approach consists in using better hardware but PLSs already push current hardware to their limits [104]. The second approach, consists in improving current simulations technologies towards efficient events processing such as better event-list algorithms [105]. Finally, the last approach proposes to improves simulations performance by using more efficient models. This is the approach used by Flow-Level Simulator (FLS), and consists in using coarse-grain network models for their efficiency property.

Instead of modeling each packet involved in communications, FLS models communications as a continuous flow of data as depicted on Figure 2.12. This has three major consequences. Firstly, since one communication generates a single event, the simulation performance are now

independent from the number of packets generated during the simulation. Instead, performance depends on the number of communications occurring during the simulation which is much lower. Secondly, considering that a communication is a continuous flow of data is an approximation and thus, leads to uncertainty regarding the communication duration. Thirdly, a continuous flow of data on a single communication channel prevents from using time multiplexing such as in real networks. Thus, two communications cannot occur at the same time. Thus, a medium resource sharing model should be used to allow for concurrent communications. Still, despite these downside, FLSs are very efficient but a close attention should be given to model validity [77]. In the remaining of this section, we will discuss about the existing FLSs to evaluate whether they can be used for end-to-end network energy consumption studies.
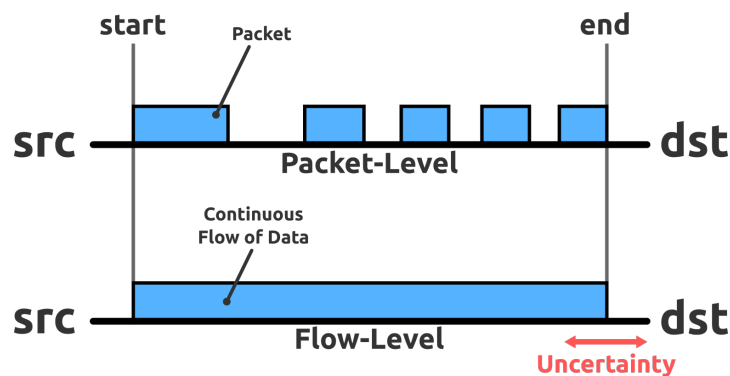
Figure 2.12: Difference between a packet-level and flow-level communication.

Narses is a FLS written in Java [59]. It proposes to simulate large-scale platforms such as peer-to-peer networks for applicative studies. Its resource sharing model is based on minimum share allocation. Narses makes the strong assumption that the Internet is following a strict hierarchical topology without internal bottlenecks. Thus, communications between a local and a distant node is only limited ether by the first or the last hop link involved in the communication. In this way, the ISP network could be ignored which improves the simulation performance. In addition, narses is definitely not designed for energy consumption studies and do not propose energy consumption models. Thus it is not suitable for end-to-end energy consumption studies.

DeSiNe is a another FLS written in C++ [106]. It is designed for application Quality of Services (QoS) and routing algorithm studies. Currently, DeSiNe do not provides any energy models which makes it unsuitable for end-to-end energy consumption studies.

FLEO is another flow-level DES based on OMNET++ [107]. FLEO proposes an efficient solution to simulate large-scale platforms such as CDN in an efficient way. Since it is a recent

Table 2.3: Flow-level network simulators review.

Simulators	Domain	Performance			Energy		
		CPU	Wired	Wireless	CPU	Wired	Wireless
Narses	WSN	✗	✓	✗	✗	✗	✗
DeSiNe	Wi-Fi	✗	✓	✗	✗	✗	✗
FLEO	Cloud	✗	✓	✗	✗	✗	✗
SimGrid	Various[a]	✓	✓	✗	✓	✗	✗

a. Can be applied to various domains.

FLS it doesn't provide as many features as the other simulators. FLEO targets only application performance evaluation. Thus, it cannot be used for end-to-end energy consumption studies.

Finally, SimGrid is a flow-level DES written in C++ [60]. The main strength of SimGrid lies in its scalability and versatility. It is used in many research domains such as Grid Computing [108], Cloud Computing [109], Fog Computing [110] and HPC [111]. It proposes a CPU model that support for applications times execution prediction for simple application but also more complex distributed applications such as Message Passing Interface (MPI) [112]. This CPU performance model comes along with a CPU energy model based on the CPU load. Additionally, SimGrid offers a VM simulation environment [113] which can be used for Cloud platforms. Regarding network communications, SimGrid gives different coarse-grain network TCP models for wired communitations. Additional bindings with the ns-3 PLS allows for easily trigger ether flow-level or packet-level simulations. Nevertheless, SimGrid lack for network energy models such as wired and wireless. Thus, SimGrid is not currently suitable for end-to-end network energy studies. However, SimGrid does provide many features compared to the other FLSs tackled in this section. This makes it a good candidate for end-to-end energy simulations.

FLS are a very interesting alternative to PLS. They propose scalable network simulations by means of coarse-grained models. Nevertheless, current state of the art FLS do not provides the required features for end-to-end energy consumption studies as shown in Table 2.3. Most of current FLS are directed towards the applicative side and lack for energy models. Consequently, to achieve end-to-end energy consumption study using a single simulation framework, we propose to extends the SimGrid FLS to benefits from its scalability, versatility (not tied to a particular use case), its CPU performance and energy models and finally its VMs model for Cloud simulations.

2.4 Conclusion

This section presented the state of the art concerning the platforms which are part of today's Internet. This taxonomy reveal that modern network platforms are composed of many nodes with a variety of hardware characteristics. Some of these platforms are composed of nodes with high computational power and data storage capabilities while some others are essentially sensors with low-power requirements. Similarly, this diversity of characteristics (or heterogeneity) is correlated to their distance to the end user. Despite this diversity, all these platforms have a common point: They are large-scale. To better understand these platforms, we presented concrete applications in which they are involved. These use cases revealed significant diversity in terms of network footprint. Thus combining this heterogeneity in terms of hardware and network footprint raise multiple challenges related to security, scalability and above all energy consumption. In fact, we showed that connecting numerous nodes together have a major impact on the energy consumption. Thus, to improve the energy efficiency of those platforms, scientists need to study them experimentally. But this task is very difficult because of their scale.

Consequently, we choose to use the simulation approach to study large-scale platforms energy consumption. This approach has several benefits such as time, money saving and reproducibility. However, network energy simulations requires to use two types models: performance and energy. The performance model allows to estimate the duration of a given task while the energy model estimates its energy consumption. Thus, combining performance and energy models in a single simulation framework could be a good solution to study the end-to-end network energy consumption starting from the IoT up to the Cloud.

However, current existing network simulators do not provide a convenient solution for end-to-end network energy consumption studies. The first category of network simulator called packet-level, suffers from performance issues related to large-scale platforms and high bandwidth applications. The other category, named flow-level, suffers from a lack of features. Thus, current state of the art related to network simulators do not offer a framework for end-to-end network energy studies. To this end, we proposed to extends a FLS called SimGrid with the required models (from Table 2.3) to provide this framework. First, in Section 4 we propose an efficient wired network energy model for flow-level simulators. Then, in Section 5 we propose the first Wi-Fi performance model designed for flow-level simulators with the aim of building a Wi-Fi network energy model as a future direction. All the models are thus implemented into SimGrid towards a unique simulation framework to study the end-to-end network energy consumption.

"If a simulator already does what you want it to do, there's a good chance you aren't asking the right questions."

—— Christos Kozyrakis [114]

AN END-TO-END ENERGY CONSUMPTION STUDY

Contents

Current network simulators face scalability and versatility issues regarding modern large-scale platforms. However, conducting end-to-end network experiments is still possible. In this chapter, we propose an end-to-end energy consumption study involving IoT, Internet Service Provider and cloud platforms on a low bandwidth use case. It uses current state of the art tools to conduct the experiments. This study aims to demonstrate that existing tools raised pratical issues while trying to study end-to-end network platforms. Additionnaly, we derive an end-to-end energy consumption model that can be used to assess the consumption of other IoT devices.

The chapter is organized as follows. Section 3.1 presents the context of the work. The low-bandwidth IoT application is characterized in Section 3.2. Details on its architecture are provided in Section 3.3. Section 3.4 provides our experimental results combining real measurements and simulations. Section 3.5 discusses the key findings of the end-to-end energy model. Finally, Section 3.6 concludes the study.

3.1 Context

Some IoT devices such as smart vehicles produce a lot of data while many others such as smart meters generate only a small amount of data. Heven so, the scale matters: many small devices can produce big data volumes. As an example, according to a report published by Sandvine in October 2018 [115], the Google Nest Thermostat is the most significant IoT device in terms of worldwide connections. It represents 0.16% of all connections and it is ranked 55th on the list of the worlwide connections. As a comparison, the voice assistants Alexa and Siri are respectively 97th and 102nd with 0.05% of all connections [115]. This example highlight the growing importance of low-bandwidth IoT applications on the Internet infrastructures, and consequently on their energy consumption.

In this chapter, we focus on low-bandwidth applications related to IoT devices such as smart meters or smart sensors. These devices send few data periodically to cloud servers, either to store them or to get computing power and take decisions. This is a first step towards a comprehensive characterization of the global IoT energy footprint. While few studies address the energy consumption of high-bandwidth IoT applications [116], to the best of our knowledge, none of them targets low-bandwidth applications, despite their growing importance on the Internet infrastructures.

Low-bandwidth IoT applications such as the Nest Thermostat often rely on sensors powered by batteries. For this type of sensors, reducing their energy consumption is a critical target. Yet, we argue that end-to-end energy models are required to estimate the overall impact of IoT devices, and to understand how to reduce their complete energy consumption. Without such models, one could optimize the consumption of on-battery devices at a heavier cost for cloud servers and networking infrastructures, resulting on an higher overall energy consumption. Using end-to-end models could prevent these unwanted effects.

This chapter aims at evaluating the existing experimental tools for end-to-end network energy consumption study. Still, it also yields the following contributions:

- A characterization of low-bandwidth IoT applications.

- An energy consumption analysis of a low-bandwidth IoT application, including the energy consumption of the Wi-Fi IoT device and the consumption induced by its utilization on the Cloud and telecommunication infrastructures.

- An end-to-end energy model for low-bandwidth IoT applications relying on Wi-Fi devices.

3.2 Characterization of low-bandwidth IoT applications

In this section, we detail the characteristics of the considered IoT application. While the derived model is more generic, we focus on a given application to obtain a precise use-case with accurate power consumption measurements.

The Google Nest Thermostat relies on five sensors: temperature, humidity, near-field activity, far-field activity and ambient light [117]. Periodical measurements, sent through wireless communications on the Internet, are stored on Google data centers and processed to learn the home inhabitants habits. The learned behavior is employed to automatically adjust the home temperature managed by heating and cooling systems.

(a) Faithful view of the IoT network architecture.

(b) Abstract view of the IoT architecture.
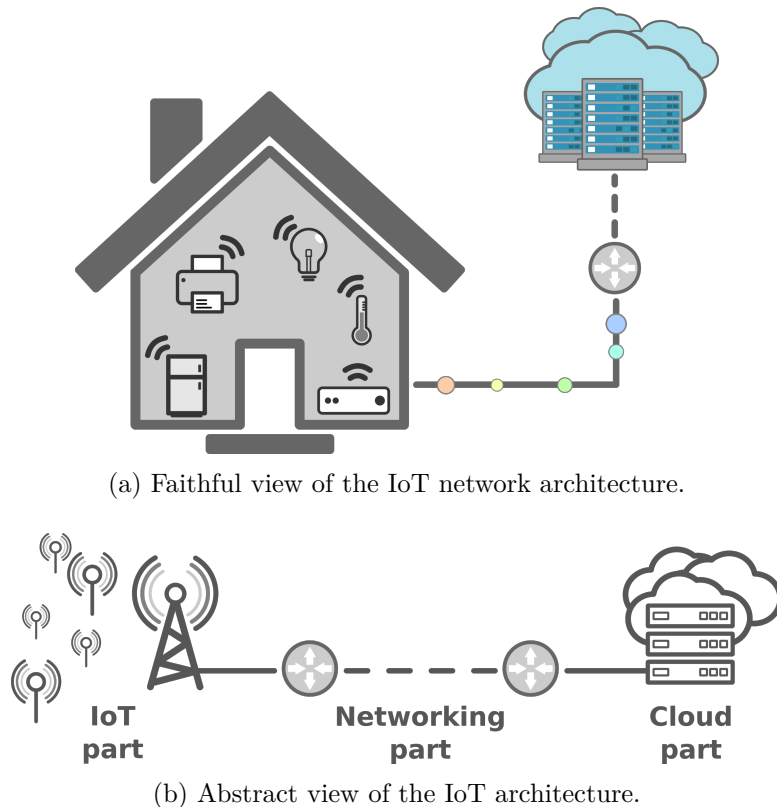
Figure 3.1: Overview of the complete IoT network architecture.

Each IoT device senses periodically its environment. Then, it sends the produced data through Wi-Fi (in our context) to its gateway or AP. The AP is in charge of transmitting the data to the cloud using the Internet. Figure 3.1a illustrates this architecture. In a home, several IoT devices can share the same AP. We consider low-bandwidth applications where devices produce several network packets during each sensing period. The transmitting frequency can

vary from one to several packet per minute [118].

We consider that the link between the AP and the Cloud is composed of several network switches and routers using Ethernet as shown in Figure 3.1b. The number of routers on the path depends on the location of the server, either in a Cloud data center or in a Fog site at the edge of the network.

We assume that the server hosting the application data for the users belongs to a shared cloud facility with classical Service Level Agreement (SLA). The facility provides redundant storage and computing means as VM. A server can host several VMs at the same time.

3.3 Experimental setup

The used experimental setup aims at evaluating the limits of the available tools in terms of end-to-end energy consumption study. The IoT and the network parts are modeled through ns-3 simulations which is a well known simulator used for network studies and detailed in Section 2.3.3. However, since no simulator are versatile enough to include Cloud simulations, we have to add an additional layer of experiment. Thus, to model the Cloud part we used real servers connected to wattmeters. By combining these tools, it is possible to evaluate the end-to-end energy consumption of the whole system.

3.3.1 IoT Part

In the first place, the IoT part is composed of several sensors connected to an AP which form a cell. In the experimental scenario, we setup between 5 and 15 sensors connected to the AP using Wi-Fi 5GHz 802.11n. The nodes are placed randomly in a rectangle of $400m^2$ around the AP which corresponds to a typical use case for a home environment. All the cell nodes employ the default Wi-Fi energy model provided by ns-3. The different energy values used by the energy model are provided in Table 3.1. These parameters were extracted from previous work [119, 116] on IEEE 802.11n. Besides, we suppose that the energy source of each node is not limited during the experiments. Thus each node can communicate until the end of all the simulations.

As a scenario, sensors send 192 bits packets to the AP. These packets are composed of: **1)** A 128 bits sensors id **2)** A 32 bits integer representing the temperature **3)** An integer timestamp representing the temperature sensing date. They are stored as time series. The data are transmitted immediately at each sensing interval I that we vary from 1s to 60s. Finally, the

Table 3.1: Simulations Energy Parameters.

(a) IoT part.		(b) Network part.	
Parameter	Value	Parameter	Value
Supply Voltage	3.3V	Idle	0.00001W
Tx	0.38A	Bytes (Tx/Rx)	3.4nJ
Rx	0.313A	Pkt (Tx/Rx)	192.0nJ
Idle	0.273A		

AP is in charge of relaying data to the cloud via the network part.

3.3.2 Network Part

The network part represents the a network section starting from the AP to the Cloud excluding the server as depicted on Figure 3.1b. We consider the server to be 9 hops away from the AP with a typical round-trip latency of 100ms from the AP to the server [116]. Each node from the AP to the Cloud is a network switch with static and dynamic network energy consumption. The first 8 hops are edge switches and the last one is considered to be a core router as mentioned in [5].

We leverage the ECOFEN energy model of ns-3, that is presented in Section 2.3.3. The Table 3.1 shows the differents parameters used to instantiate the ECOFEN energy model. These values were extracted from previous works [101].

3.3.3 Cloud Part

Finally, to measure the energy consumed by the Cloud part, we use a real server from the large-scale test-bed Grid'5000 describe in Section 2.1.1. Grid'5000 provides clusters composed of several nodes which are connected to wattmeters. The wattmeters provide 50 instantaneous power measurements per second and per server. This way, we can benefit from real energy measurements. The server used in the experiment embeds two Intel Xeon E5-2620 v4 processors with 64 GB of RAM and 600GB of disk space on a Linux based operating system. This server is configured to use KVM as virtualization mechanism. We deploy a classical Debian x86_64 distribution on the VM along with a MySQL database. We use different amounts of allocated memory for the VM namely 1024MB/2048MB/4096MB to highlight its effects on the server energy consumption. The server only hosts this VM to easily isolate its power consumption.
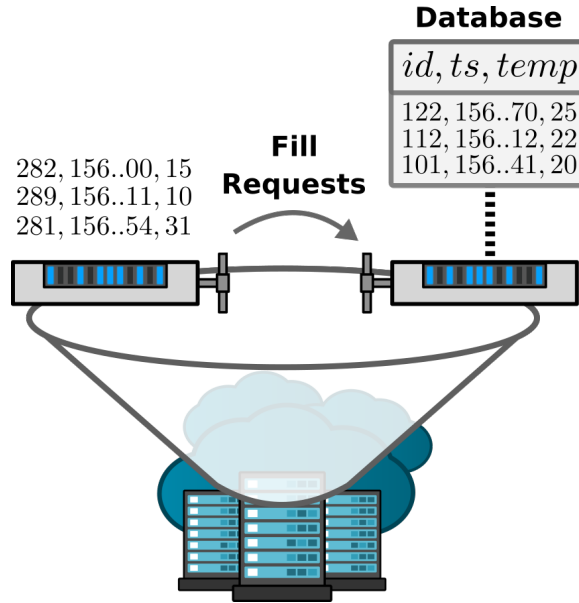
Figure 3.2: Grid'5000 experimental setup.

One drawback of using this second experimentation process for the Cloud part is that the IoT devices cannot directly communicate with the server. Thus, the data sent by the IoT devices are simulated using another server from the same cluster. This server is in charge of sending the data packets to the VM hosting the application to fill its database. In the following, each data packet coming from an IoT device and addressed to the application VM is called a request. Consequently, it is easy to vary the different application characteristics namely: **1)** The number of requests, to virtually add/remove sensors **2)** The requests interval, to study the impact of the transmitting frequency. Figure 3.2 presents this experimentation setup.

3.4 Evaluation

3.4.1 IoT and Network Power Consumption

In this section, we analyze the experimental results. We first study the impact of the sensors' transmission frequency on their energy consumption. To this end, we run several simulations in ns-3 with 15 sensors using different transmission frequencies. The results are shown on Table 3.2.

These results show that the transmission frequency has a very low impact on the energy consumption and on the average end-to-end application delay. The impact exists, but remains very limited. This due to the fact that in such a scenario with very small number of communications

Table 3.2: Sensors transmission interval effects with 15 sensors.

Transmission Interval	10s	30s	50s	70s	90s
Sensor Power	13.51794W	13.51767W	13.51767W	13.51767W	13.51761W
Network Power	0.44188W	0.44177W	0.44171W	0.44171W	0.44171W
Application Delay	0.09951s	0.10021s	0.10100s	0.10203s	0.10202s
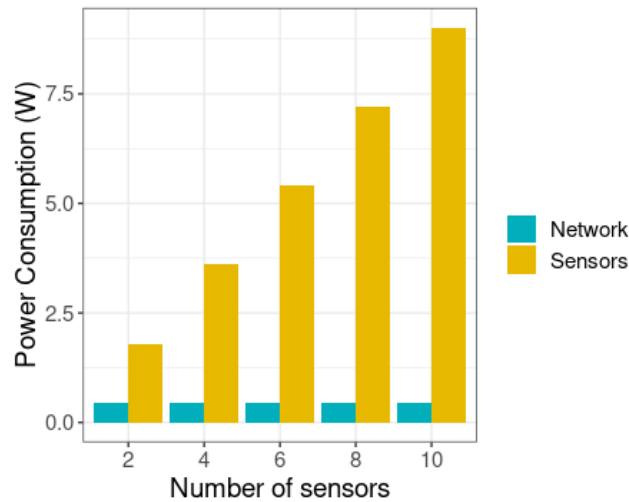
Figure 3.3: Analysis of the variation of the number of sensors on the IoT/Network part energy consumption for a transmission interval of 10s.

spread over the time, the contention of the Wi-Fi channel remains very low.

Previous work [116] on a similar scenario shows that increasing application accuracy impacts strongly the energy consumption in the context of data stream analysis. In our case, application accuracy is driven by the sensing interval and thus, the transmission frequency of the sensors. It is characterized by small and sporadic network traffic. Results show that with a reasonable transmission interval, the energy consumption of the IoT and the network parts are almost not affected by the variation of this transmission interval. In fact, transmitted data are not large enough to leverage the energy consumed by the network.

We then vary the number of sensors in the WiFi cell. Figure 3.3 represents the energy consumed by the sensor and the network (from the AP to the cloud) parts according to the number of sensors. Similarly to the results of Table 3.2, the network part is almost not affected by the number of sensors as their traffic is negligible compared to the network devices capacities. Consequently, sensors energy consumption is dominant, as each sensor adds its own consumption.
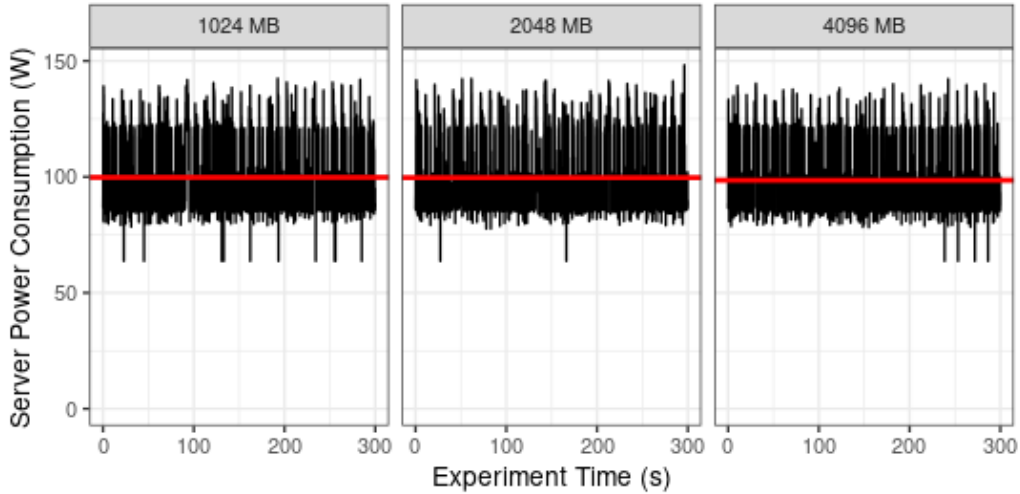
Figure 3.4: Server power consumption multiplied by the PUE (= 1.2) using 20 sensors sending data every 10s for various VM memory sizes.

3.4.2 Cloud Energy Consumption

In this end-to-end energy consumption study, cloud accounts for a huge part of the overall energy consumption. According a report [120] on United States data center energy usage, the average PUE of an hyper-scale data center is 1.2. Thus, all our energy measurements on the cloud server will account for the PUE in our analysis. It means that the power consumption of the server is multiplied by the PUE to include the external energy costs such as server cooling and data center facilities [121].

Firstly, we analyze the impact of the memory allocated by the VM on the server energy consumption. Figure 3.4 depicts the server energy consumption according to the memory allocated by the VM for 20 sensors sending data every 10s. Note that the horizontal red line represents the average energy consumption for the considered sample of energy values. We can see that at each transmission interval, the server faces to spikes of energy consumption. However, the amount of allocated memory to the VM does not significantly influence the server energy consumption. In fact, simple database requests do not need any particular heavy memory accesses and processing time. Thus, the remaining experiments are solely based on VM with 1024MB of allocated memory.

Next, we study the effects of increasing the number of sensors on the server energy consumption. Figure 3.5a presents the results of the average server energy consumption when varying the number of sensors from 20 to 500. Figure 3.5b presents the average server energy cost per sensor according to the number of sensors. These results show a clear linear relation between
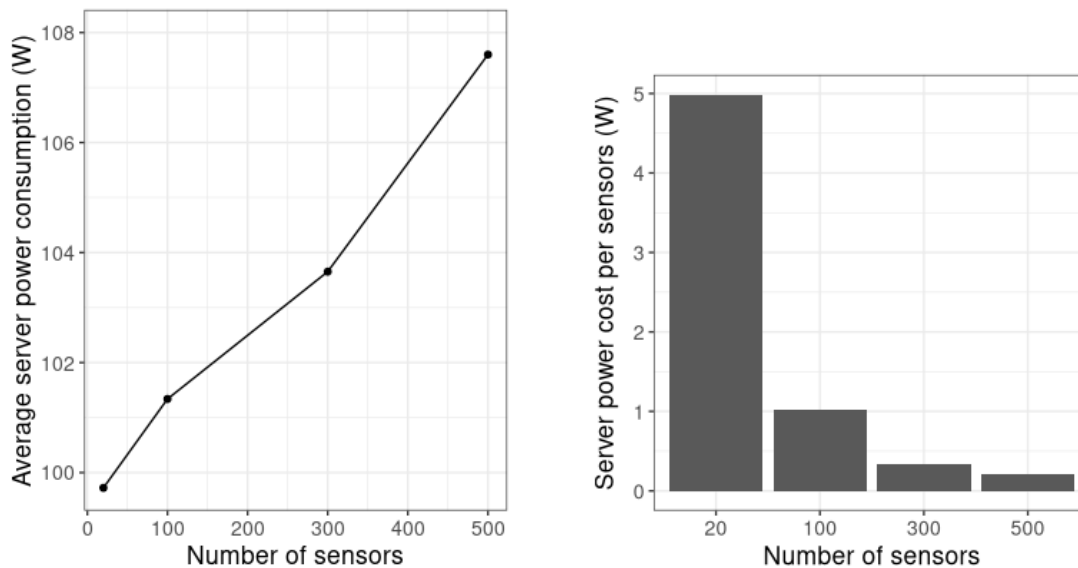
(a) Average server energy consumption multiplied by the PUE (= 1.2).

(b) Average sensors energy cost on the server hosting only our VM.

Figure 3.5: Server energy consumption multiplied by the PUE (= 1.2) for sensors sending data every 10s.

the number of sensors and the server energy consumption. Moreover, we can see that the more sensors we have per VM, the more energy we can save. In fact, since the server's idle power consumption is high (around 97 Watts), it is more energy efficient to maximize the number of sensors per server. As shown on Figure 3.5b, a significant amount of energy can be save when passing from 20 to 300 sensors per VM. Note that these measurements are not the row measurements taken from the wattmeters: they include the PUE but they are not shared among all the VMs that could be hosted on this server. So, for the studied server, its static power consumption (also called idle consumption) is around 83.2 Watts and we consider a PUE of 1.2 (this value is taken from [120]).

A last parameter can leverage server energy consumption, namely sensors transmission interval. In addition to increasing the application accuracy, sensors transmission frequency increases network traffic and database accesses. Figure 3.6 presents the impact on the server energy consumption when changing the transmission interval of 50 sensors to 1s, 10s and 30s. We can see that, the lower the sensors transmission interval is, the more server energy consumption peaks ($\approx$ 150W) occur. Therefore, it leads to an increase of the server energy consumption. An end-to-end analysis is mandated to fully understand this tradeoff.
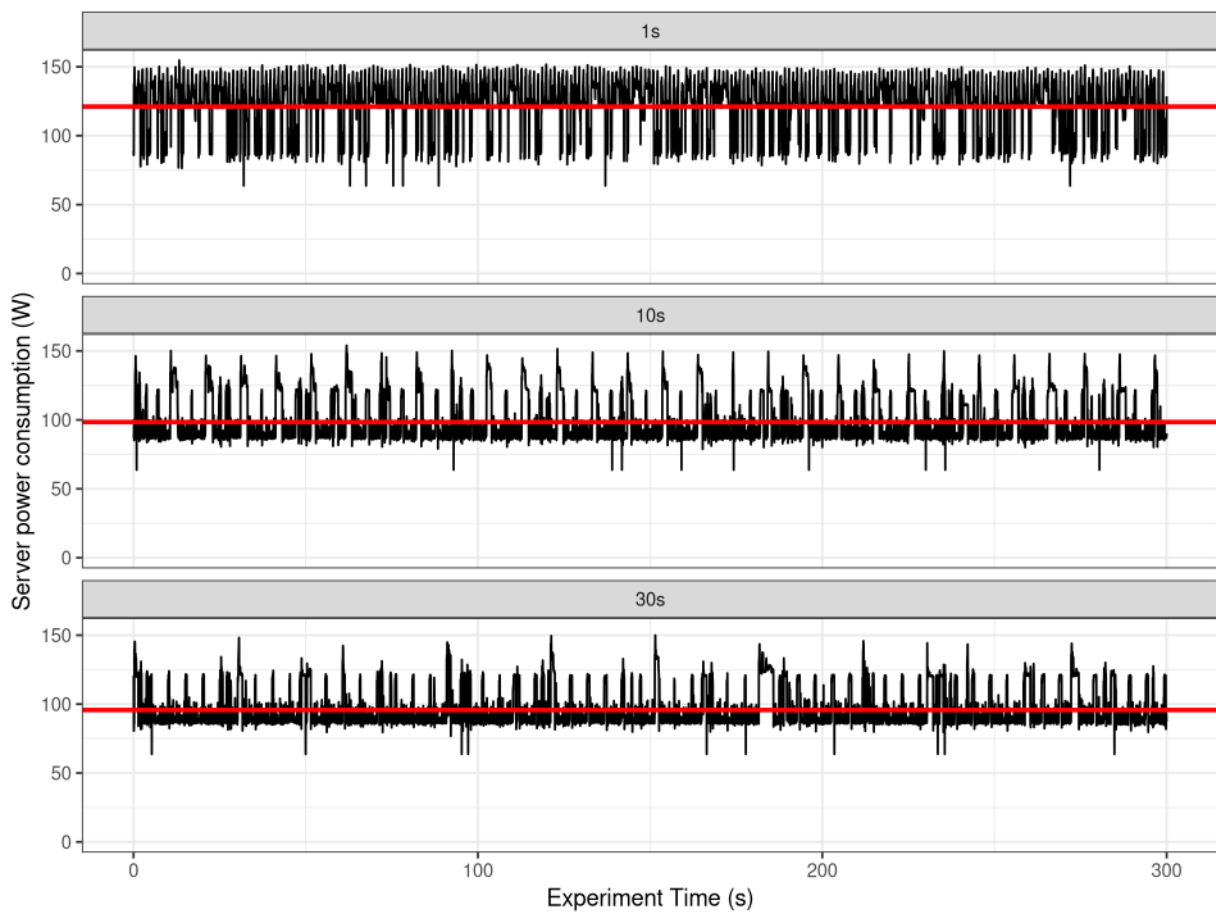
Figure 3.6: Server energy consumption multiplied by the PUE (= 1.2) for 50 sensors sending requests at different transmission interval.

3.5   End-to-end Consumption Model

To have an overview of the energy consumed by the overall system, it is important to consider the end-to-end energy consumption. We detail here the model used to attribute the energy consumption of our application for each part of the architecture:

1. For the IoT part, the entire consumption of the IoT device belongs to the system's accounted consumption.

2. For the network part, the data packets generated by the IoT device travel through network switches, routers and ports that are shared with other traffic.

3. For the cloud part, the VM hosting the data is shared with other IoT devices belonging to the same application and the server hosting the VM also hosts other VMs. Furthermore, the server belongs to a data center and takes part in the overall energy drawn to cool the server room.

Concerning the IoT part, we include the entire IoT device power consumption. Indeed, in our targeted low-bandwidth IoT application, the sensor is dedicated to this application. From Table 3.1, one can derive that the static power consumption of one IoT sensor is around 0.9 Watts. Its dynamic part depends on the transmission frequency.

Concerning the sharing of the network costs, for each router, we consider its aggregate bandwidth (on all the ports), its average link utilization and the share taken by our IoT application. For a given network device, we compute our share of the static energy part as follows:

$$P_{static}^{netdevice} = \frac{P_{static}^{device} \times Bandwidth^{application}}{AggregateBandwidth^{device} \times LinkUtilization^{device}}$$

where P_{static}^{device} is the static power consumption of the network device (switch fabrics for instance or gateway), $Bandwidth^{application}$ Is the bandwidth used by our IoT application, $AggregateBandwidth^{device}$ is the overall aggregated bandwidth of the network device on all its ports, and $LinkUtilization^{device}$ is the effective link utilization percentage. The $Bandwidth^{application}$ depends on the transmission frequency in our use-case. The formula includes the link utilization to charge for the effective energy cost per traffic and not for the theoretical upper bound which is the link bandwidth. Indeed, using such an upper bound leads to greatly decrease our energy part, since link utilization typically varies between 5 to 40% [122, 123].

Similarly, for each network port, we compute the share attributable to our application: the ratio of our bandwidth utilization over the port bandwidth multiplied by the link utilization

Table 3.3: Network Devices Parameters.

Device	Parameters
Gateway	Static power = 8.3 Watts, Bandwidth = 54Mbps, Utilization = 10%
Core router	Static power = 555 Watts, 48 ports of 1 Gbps, Utilization = 25%
Edge switch	Static power = 150 Watts, 48 ports of 1 Gbps, Utilization = 25%

and the overall static power consumption of the port. Table 3.3 summarizes the parameters used in our model, they are taken from [124, 122]. These parameters are used in our formula to compute the values used in the simulations. They are presented in left part of Table 3.1.

Concerning the Cloud costs, we take into account the number of VMs that a server can host, the CPU utilization of a VM and the PUE. For a given Cloud server hosting our IoT application, we compute our share of the static energy part as follows:

$$P_{static}^{Cloudserver} = \frac{P_{static}^{server} \times PUE^{DataCenter}}{HostedVMs^{server}}$$

Where P_{static}^{server} is the static power consumption of the server, $PUE^{DataCenter}$ is the data center PUE, and $HostedVMs^{server}$ is the number of VMs a server can host. This last parameter should be adjusted in the case of VMs with multiple virtual CPUs. We do not consider here over-commitment of Cloud servers. Yet, the dynamic energy part is computed with the real dynamic measurements, so it accounts for VM over-provisioning and resource under-utilization.

In our case, the Cloud server has 16 cores, which corresponds to the potential hosting of 16 small VMs with one virtual CPU each, with each vCPU pinned to a server core. We consider that for fault-tolerance purpose, the IoT application has a replication factor of 2, meaning that two cloud servers store the database.

The Figure 3.7 represents the end-to-end system energy consumption using the model described above while varying the number of sensors for a transmission interval of 10 seconds. The values are extracted from the experiments presented in the previous section.

Note that, for small-scale systems, with Wi-Fi IoT devices, the IoT sensor part is dominant in the overall energy consumption. Indeed, the IoT application induces a very small cost on Cloud and network infrastructures compared to the IoT device cost. But, our model assumes that a single VM is handling multiple users (up to 300 sensors), thus being energy-efficient. Conclusions would be different with one VM per user in the case of no over-commitment on the Cloud side. For the network infrastructure, in our case of low-bandwidth utilization (one data packet every 10 seconds), the impact is almost negligible.
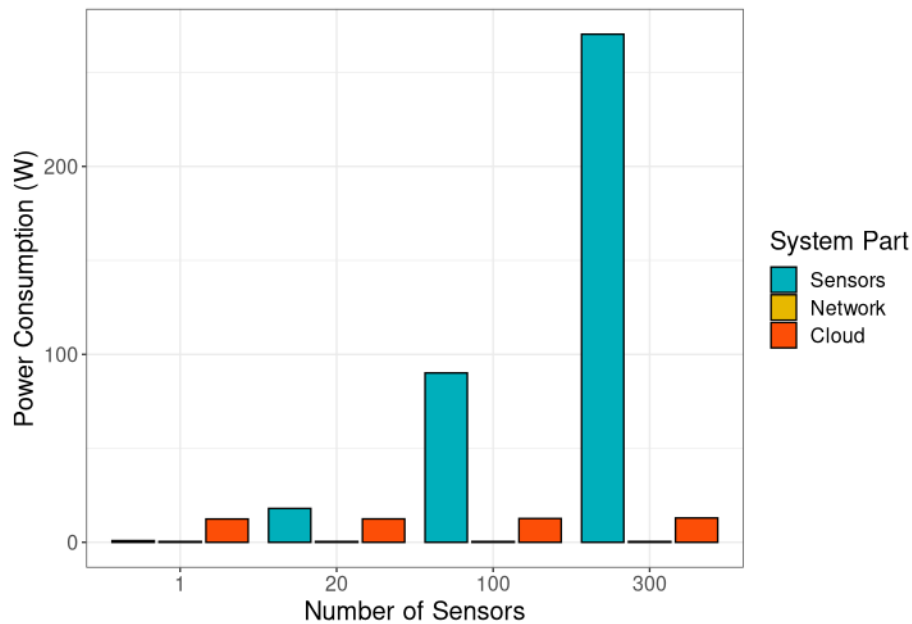
Figure 3.7: End-to-end network energy consumption using sensors interval of 10s.

Another way of looking at these results is to observe that only for a high number of sensors (more than 300), the power consumption of Cloud and network parts start to be negligible (few percent). It means that, if IoT applications handle clients one by one (i.e. one VM per client), the impact is high on cloud and network part if they have only few sensors. The energy efficiency is really poor for only few devices: with 20 IoT sensors, the overall energy cost to handle these devices is almost doubled compared to the energy consumption of the IoT devices themselves. Instead of increasing the number of sensors, which would result in a higher overall energy consumption, one should focus on reducing the energy consumption of IoT devices, especially Wi-Fi devices which are common due to Wi-Fi availability everywhere. One could also focus on improving the energy cost of Cloud and network infrastructure for low-bandwidth applications and few devices.

3.6 Conclusion

The presented experiment combines simulations and real measurements to study the energy impact of IoT devices. In particular, we analyze the energy consumption of Cloud and communication infrastructures induced by the utilization of connected devices. Through the fine-grained analysis of a given low-bandwidth IoT device periodically sending data to a Cloud server using

Wi-Fi, we propose an end-to-end energy consumption model. This model provides insights on the hidden part of the iceberg: the impact of IoT devices on the energy consumption of Cloud and network infrastructures. On our use-case, we show that for a given sensor, its larger energy consumption is on the sensor part. But the impact on the Cloud and network part is huge when using only few sensors with low-bandwidth applications. The simulations and the experiments presented in this chapter are available online [125].

This experiment is also interesting on the methodological aspect, because it highlights the difficulties faced for such studies. First, we had to split our experimentation in two part to model the energy consumption of the whole system. Consequently, we had to simulate the IoT traffic on the servers side. Moreover, the fact that we did not used a unified framework leads to a lack of experimental flexibility. Indeed, we were not able to easily integrate several IoT platforms along with multiple servers connected to a common ISP network. The lack of scalability on the simulation part prevents us from simulating more than 50 devices in a resonable amount of time. Thus, we had to extrapolate the data to obtain energy consumption predictions up to 300 hundred sensors. Then, the fact that we had to simulate the IoT traffic on the server part add more complexity for horizontal scaling. The following chapters will alleviate some of these difficulties.

EFFICIENT WIRED NETWORK ENERGY MODEL

Contents

Chapter 2 and 3, demonstrate that the use of simulation is a good way for scientists to develop, improve and assess models that predict the network energy consumption. However, as stated in Section 2.3.3, packet-level simulators start to reach their limits in terms of performance. This calls for a new solution to study the wired network energy consumption of large-scale platforms. In this chapter, we propose two energy models for wired networks adapted to flow level simulation to estimate the energy consumption of large-scale platforms. An evaluation of these models is proposed to demonstrate their applicability and their accuracy.

This chapter is structured as follows. Section 4.1 introduces the context of the contribution along with the problem. Section 4.2 presents the proposed energy models. Section 4.3 details our evaluation methodology. Then, an evaluation of the energy models in terms of validity, scalability and applicability is proposed in Section 4.4. Finally, we conclude in Section 4.5.

4.1 Context and Problem

Current network platforms are getting larger. The Section 2.1.3 details the different challenges and use cases raised by these platforms. In particular, we are facing to an increase of the energy consumed by wired network platforms such as data centers and ISP networks. To study and improve their energy efficiency, network simulators can be used as a experimental framework. In addition, providing a scalable solution for wired network energy simulation pushes a step towards the end-to-end network energy simulation as presented in Figure 1.2.

The energy consumption of wired networks can be estimated by several types of simulators. However, they work at a fine grain – simulating each packet exchange – and thus are not scalable enough for modern research purposes which involves large network platforms and potentially huge network traffic volumes. Indeed, their fine-grained network models are not compatible with efficient simulations. Nevertheless, scalable simulators such as FLSs exist but to the best of our knowledge, none of them provide network energy consumption models while not being limited to a specific network domain (such as Cloud or Fog). FLSs operate at a communication flow level and do not simulate each packet. Thus, this approach allows to reduce simulation execution time and memory footprint. In practice, FLSs constitute a nice compromise between network abstraction and scalability. But their coarse-grained nature makes current wired network energy model not directly applicable to FLSs.

Therefore, to the best of our knowledge there is no efficient network simulator with a scalable wired energy model to study the energy consumption of large-scale platforms. Consequently, in this work we propose to extend the existing flow-based simulator SimGrid with a wired network energy model. We show that modeling the network energy consumption into a flow-level simulator leads to predictions that are in line with the predictions of packet-level simulators. We also show that simple linear models are sufficient on realistic settings to obtain accurate values, while more complex models were proposed in literature. Additionally, we will demonstrate the applicability of our models on large-scale platforms.

4.2 Contribution

4.2.1 A Link based Wired Network Energy Model

Simulators are able to abstract the complexity of network devices energy consumption. As explained in Section 2.2.2, even if a network equipment is not subject to any network traffic, it consumes energy. Indeed, the different components involved in network devices should be

maintained powered on to ensure a fully functional network communication in case of packet arrival. This energy consumption part, called the idle or static part can represent 85% of the device peak power [126]. Classically, the energy consumption is estimated using the energy definition presented in the Equation 2.1. Considering a network device, its energy consumption varies over time according to the network traffic. Since its power is composed of two parts: one static and one dynamic, we can define its overall power consumption such as in the Equation 2.3. Existing energy models in the literature are based on this equation that involves energy cost per network packet. Thus, they are not directly compatible with flow-level simulators. Indeed, flow-level simulators do not provide any notion of port in their models nor packets. These energy models should be adapted to this network representation.

SimGrid does not include Network Interface Controllers (NICs) and routers. Instead, the paths between two hosts are represented by routes. Each route is composed of several entities, called links that represent both, the wire and the NICs at each end. A route between two hosts can be composed of multiple links but not necessarily of multiple hosts. To this end, there is no notion of port in flow-level simulators. In spite of this, we can attach the energy consumption of ports on the links. Thus, we propose to define the instantaneous power of the overall network platform using the following equation:

$$P_{total}(t) = \sum_{i \in L} \left[P^i_{static} + P^i_{dynamic} \times \text{Load}_i(t) \right] \tag{4.1}$$

with L representing the set of links on the simulated network platform, P^i_{static} the static power of the link i, $P^i_{dynamic}$ the dynamic power of the link i and Load_i the load (usage) of the link i. As stated previously, wired network energy models are using low-level instantiation parameters such as the energy consumption per byte and per packets which are not available in FLS. Hence, we propose to use a wired network energy model using a linear equation with an intercept equal to the idle port power consumption and a maximum power consumption defined as follow:

$$P_{max} = P_{idle} + \underbrace{BW \times \left[E_{ByteCons} + \frac{E_{PktCons}}{MTU} \right]}_{\text{Dynamic Part}} \tag{4.2}$$

In this equation, BW represents the maximum port rate in Bps, $E_{ByteCons}$ its energy consumption per byte in Joules, $E_{PktCons}$ represents the network device energy consumption needed for handling a packet in Joules. Finally, the MTU (Maximum Transmission Unit) is used as an over-approximation of the packet size. Given this equation, we are able to estimate the

energy consumed by a single port using a link. However, in every real networks, each link is connected to two different ports with potentially different energy consumption schemes. This represents another difficulty for transposing packet-level energy models into FLSs due to their coarse-grain nature. To solve this issue, we propose to define two energy models based on the Equation 4.2. The first one is dedicated to platforms with ports that have the same energy consumption characteristics (called homogeneous energy platforms). The second model is dedicated to heterogeneous energy platforms.

4.2.2   Homogeneous model for flow-based simulators

Firstly, we introduce an energy model for network topologies that uses homogeneous network devices. As said previously, links will hold the energy consumption for each of their ports. In addition, links should ensure the bandwidth sharing. To achieve this goal in SimGrid, we propose to use one split-duplex link between each end-node. In SimGrid, a split duplex link creates two physicals links. One is dedicated to the upward traffic and the other one for the backward traffic. Thus, each of these two links carries an energy consumption model (one for each port). However sending to an upward or backward link should involve the energy consumption of 2 ports namely the source and the destination. Thus, to overcome the lack of port representation into SimGrid, we double the dynamic energy consumption of the upward and backward link to account for the energy consumption of the two ports (one at each end of the link). This operation is possible since both ports have the same energy consumption characteristics (homogeneous platform). Figure 4.1 depicts this homogeneous energy model for both PLS and FLS. Using the Equations 4.1 and 4.2 the port power consumption of the overall platform at any time t can be express as follow:

$$P_{total}(t) = \sum_{i \in L} \left[P_{idle} + 2 \times BW_i \times (E_{ByteCons} + \frac{E_{PktCons}}{MTU}) \times \text{Load}_i(t) \right] \qquad (4.3)$$

This energy model has multiple advantages. First, it is very easy to instantiate because there is only one energy cost per link. Consequently, this makes it easier to implement on flow-level simulators since these simulators use at least links for communications (but not necessarily ports). Finally, this model provides low computational overhead. However, many real topologies use heterogeneous networking devices. Thus, the energy consumption from one NIC to another varies according to its specifications. Consequently, this heterogeneity cannot be explicitly represented using the homogeneous model. Hence, we decide to introduce another model to overcome this limitation.
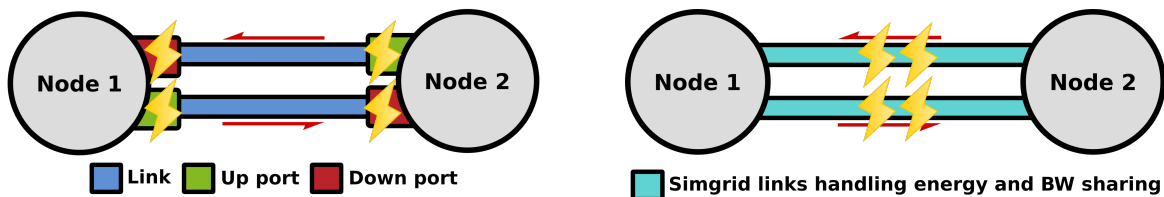
(a) Homogeneous model on packet level simulators.

(b) Homogeneous model on flow level simulators.

Figure 4.1: Homogeneous model on packet/flow level simulators.

4.2.3 Heterogeneous model

In SimGrid, routes from one host to another can be composed of multiple links. Therefore, we can take advantage of this feature to build the heterogeneous model. To model two ports with different energy consumption values on a single link, we introduce routes made of 3 links. The first link models the energy consumption of the first port, the second link handles the bandwidth sharing and finally, the third link models the energy consumption of the second port. However, we are using split-duplex links, thus two links will be created by SimGrid and consequently the idle power will be multiplied by two. To overcome this issue, we simply divide by two the idle power of each energy link. Thus, this energy model follows a linear behavior with a minimum value equals to $idle/2$ and a maximum value defined as:

$$P_{max} = \frac{P_{idle}}{2} + \underbrace{BW \times \left[E_{ByteCons} + \frac{E_{PktCons}}{MTU} \right]}_{DynamicPart} \tag{4.4}$$

The Figure 4.2 depicts this heterogeneous energy model for both PLS and FLS. An XML platform example for the heterogeneous energy model is provided on Listing 4.1. Using the Equations 4.1 and 4.2 the port power consumption of the overall platform at any time t can be express as follow:

$$P_{total}(t) = \sum_{i \in L} \left[\frac{P_{idle_i}}{2} + BW_i \times (E_{ByteCons_i} + \frac{E_{PktCons_i}}{MTU}) \times \text{Load}_i(t) \right] \tag{4.5}$$

Now we have defined both energy models, the next step is to assess them in terms of prediction accuracy and scalability.

(a) Heterogeneous model on packet level simulators.

(b) Heterogeneous model on flow level simulators.
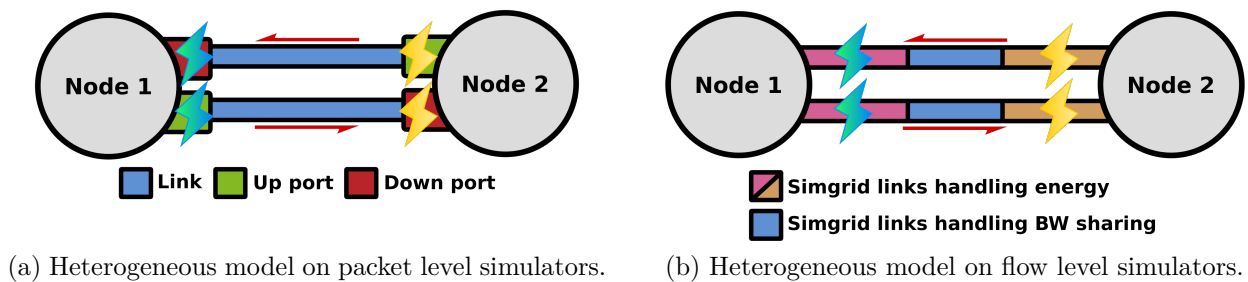
Figure 4.2: Heterogeneous model on packet/flow level simulators.

```xml
<?xml version='1.0'?>
<!DOCTYPE platform SYSTEM
    "http://simgrid.gforge.inria.fr/simgrid/simgrid.dtd">
<platform version="4.1">
    <AS id="AS0" routing="Full">
        <host id="Node0" speed="100.0Mf,50.0Mf,20.0Mf" pstate="0">
        </host>
        <host id="Node1" speed="100.0Mf,50.0Mf,20.0Mf" pstate="0">
        </host>

        <link id="Link11" bandwidth="${BWPAR}" latency="0ms"
            sharing_policy="SPLITDUPLEX">
            <prop id="watt_range" value="${NODE0-MIN}:${NODE0-MAX}" />
        </link>
        <link id="Link12" bandwidth="${BWPAR}" latency="${LATPAR}ms"
            sharing_policy="SPLITDUPLEX">
            <prop id="watt_range" value="0:0" />
        </link>
        <link id="Link13" bandwidth="${BWPAR}" latency="0ms"
            sharing_policy="SPLITDUPLEX">
            <prop id="watt_range" value="${NODE1-MIN}:${NODE1-MAX}" />
        </link>

        <route src="Node0" dst="Node1" symmetrical="NO">
            <link_ctn id="Link11_UP" />
            <link_ctn id="Link12_UP" />
            <link_ctn id="Link13_UP" />
        </route>
        <route src="Node1" dst="Node0" symmetrical="NO">
            <link_ctn id="Link11_DOWN" />
            <link_ctn id="Link12_DOWN" />
            <link_ctn id="Link13_DOWN" />
        </route>
    </AS>
</platform>
```

Listing 4.1: Example of a SimGrid XML platform involving two hosts with the heterogenous energy model.

4.3 Methodology and Experimental Setup

4.3.1 Methodology

The evaluation of the two energy models is divided in two steps. The first step is dedicated to assessing the models by means of microbenchmark simulations. Hence, the experiments are launched using small platforms to achieve simulations on a simple and controlled environment. The second step of the evaluation is focused on testing the energy models on a real large-scale scenario. In this way, we can demonstrate that the models are suitable for real research purposes and thus, are useful for the scientific community. In addition, it demonstrates that the energy models are scalable regardless of the simulated platform and the workload size. It is important to note that our evaluation focuses on the network energy consumption estimation and not on the network performance model. Yet, it is mandatory to have accurate time estimations to get accurate energy consumptions.

To conduct the validation experiments, we decided by lack of real large-scale platforms, to do simulations. In fact, validating these energy models using test-beds requires to measure the energy consumption per byte and per packet on each network device. However, these measurements require high-precision instruments. Nevertheless, these energy measurements results are proposed in the literature [62, 127]. Therefore, we based our experiments on these works.

To validate the two energy models, the experiments are done in parallel on a packet-level simulator that acts as a trusted party. We choose to use ns-3 and ECOFEN for this purpose. It is a logical choice since ECOFEN has been assessed as accurate in the literature [128]. Moreover, ns-3 is a PLS. Thus it will be useful for the scalability experiments to compare its simulation performance to SimGrid. It is important to mention that all the simulations ran on Grid'5000, a large-scale test-bed for experiment-driven research in computer science. In fact, microbenchmark scenarios require 96 unique simulations for only one experiment point (4 data transfer scenarios, 3 topologies with 2 energy instantiation, 3 models are tested including one which is an optimization). Overall it represents 1 440 simulations. Moreover, the real use case experiment is also computationally intensive for ns-3 and ECOFEN.

All the experiments presented in this work are available online [129]. We did our best to provide reproducible experiments that can be launch with different parameters and require the least possible amount of user interactions for the reader desirous of reproducing our experiments.

4.3.2 Experimental setup

To conduct microbenchmark simulations, we designed different experimental scenarios. These scenarios are built to account for the effects on the energy consumption of the following parameters: **1)** The number of nodes and their connections in the network **2)** Different flow configurations, and thus different bandwidth sharing patterns **3)** The heterogeneity of the devices in terms of energy profile. For this, we define three platforms. The first one is composed of two hosts connected by one link. The second platform is an extension of the first one where we add two hosts and two links. Finally, the last platform, called Dogbone, is composed of six hosts connected together to form a bottleneck on the two central hosts. Furthermore, the platforms can be set up with homogeneous or heterogeneous energy consuming devices.

Thanks to these two variants, the ability of each model to predict the overall energy consumption of platforms with homogeneous and heterogeneous devices is highlighted. These scenarios are summarized on Figure 4.3. Then, each platform is running with different TCP flow configurations. Other classical network parameters are also defined. These parameters are equals for all the microbenchmark scenarios: the latency is set to 10ms and the bandwidth to 1Gbps for each link since the energy value for such a NIC is available in the literature [62, 127]. Thus according to these works, the energy values used in our microbenchmarks for 1Gbps links are 1.12 Watts for idle power consumption, 3.4nJ for energy consumed per byte and 197.2nJ for the energy consumption per packet. Regarding the heterogeneous platform instantiation, we also use additional energy values for 10Gbps links namely 0.53 Watts for idle power consumption, 14nJ and 1 504nJ respectively for energy consumption per byte and per packet. All these energy parameters are summarized in Table 4.1. Finally, the data flow size is varying randomly between 10MB to 100MB (x-axis) to build a power profile of the network platforms. We used random data flow size values in order to avoid bias that could occur with evenly distributed samples.

Table 4.1: Overview the energy parameters used in the simulation.

Parameters	Ports	
	1Gbps	10Gbps
Idle Power	1.12W	0.53W
Packet Energy	197.2nJ	1 504nJ
Byte Energy	3.4nJ	14nJ
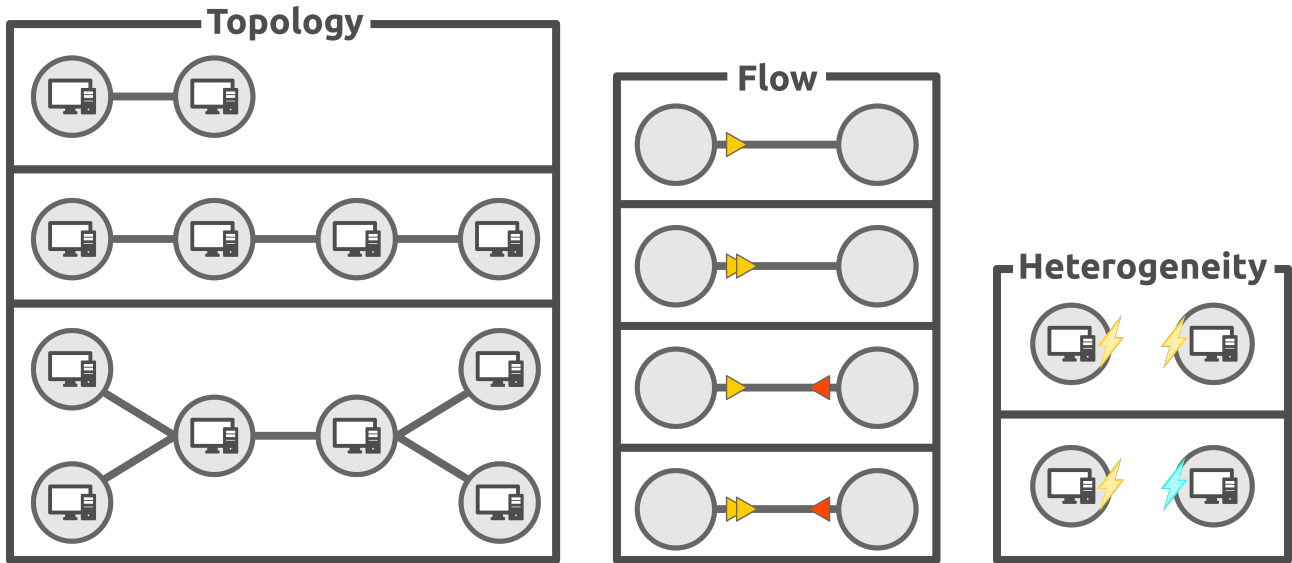
Figure 4.3: Overview of the microbenchmarks scenarios.

4.4 Evaluation: validity, scalability and real use case

4.4.1 Validity of the proposed models

As a first step, we simulate the microbenchmark scenarios on homogeneous energy consumption platforms. The amount of data sent by the hosts varies according to the parameters defined in Section 4.3.2. The results for the different platforms are shown on Figure 4.4. The flow configuration used for these scenarios is the following: 3 flows overall (2 up and 1 down) between the extreme nodes of each platform. The results for the other scenarios (not displayed) are similar.

These first results show the ability of each model to predict the overall platform' ports (Total Energy) energy consumption (static and dynamic) with homogeneous network devices. The results show a linear relation between the amount of data sent between hosts and the overall energy consumption. One interesting phenomenon to note is that both SimGrid energy models predict the exact same amount of energy. Their energy relative error is close to 0, which means they have the same energy prediction ability. In fact, it is not surprising since the first energy model is a subset of the second one. Thus, they show similar predicting abilities on homogeneous platforms. To measure the energy prediction accuracy of our models we used the Average Relative Error (ARE) metric. More specifically we based the ARE on the logarithmic
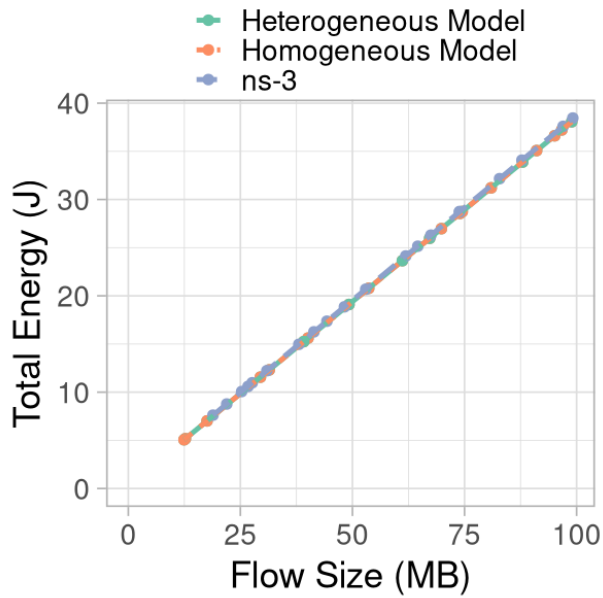
75

(a) 2 hosts 1 link platform power profile
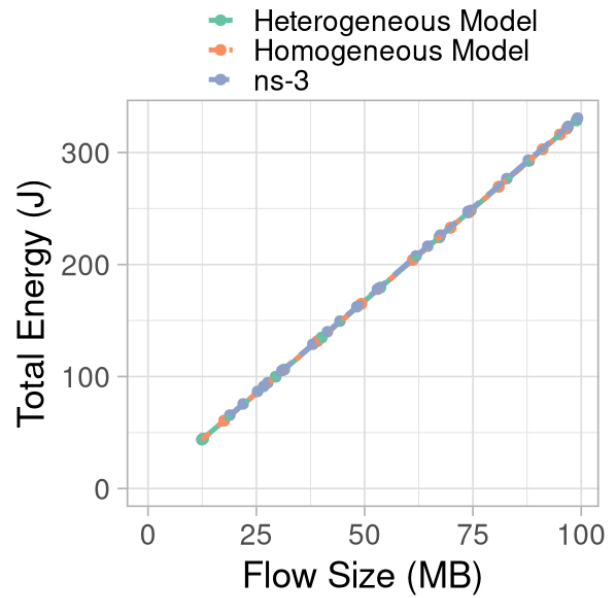
(b) 4 hosts 3 links platform power profile
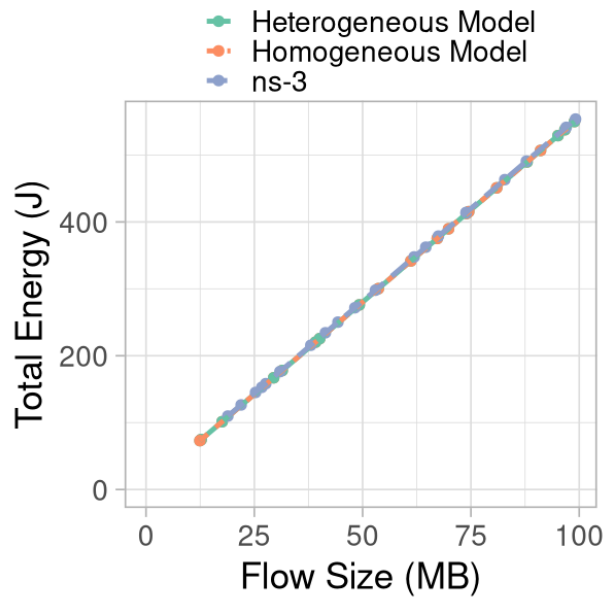
(c) Dogbone platform power profile

Figure 4.4: Microbenchmarks energy consumption on platforms with homogeneous devices.

error metric borrowed from [77]:

$$\text{MeanLogErr} = |\log_{10}\left(\frac{\hat{E}_{total}}{E_{total}}\right)| \quad \text{with MeanLogErr} \in \,]0, +\infty[\quad (4.6)$$

were $\hat{E}$ is the estimated energy consumption (SimGrid predictions) and E is the reference energy consumption (ns-3). Then the ARE can be deduced with the following formula:

$$\text{MeanRelErr} = \exp_{10}(\text{MeanLogErr}) - 1 \quad (4.7)$$

The measured ARE between both energy models (homogeneous and heterogeneous) and the ECOFEN ns-3 module shows a very high prediction accuracy with MeanRelErr < 1%.

Next, to demonstrate the ability of each model to predict the energy consumption on heterogeneous platforms, we run similar microbenchmark experiments using the scenarios with heterogeneous devices in terms of energy consumption. Thereby, we expect the homogeneous model to have a lower prediction accuracy since it is not designed to handle heterogeneous energy platforms.

The simulation results are shown on Figure 4.5. As expected, the homogeneous model is less accurate than the heterogeneous model. In fact, we can clearly see that the heterogeneous model energy estimation corresponds to the energy predicted by ns-3 and ECOFEN. However, the homogeneous model is doing wrong approximations. This phenomenon is explain by the fact that the granularity of the first energy model is not fine enough to fully capture the dynamic energy consumption of the network devices.

Nonetheless, we decided to modify the instantiation of the homogeneous model to improve its prediction capability. Hence, for each link we attach energy values that are equal to the average values between the two ports at each end of the link. The results of these simulations are also visible on Figure 4.5 under the "Optimized Homogeneous Model". Surprisingly, by using this simple optimization we observe that the homogeneous model produces very accurate results almost equal to the heterogeneous model. This improvement can be explained by the fact that on real platforms, the traffic going through each port of a link is almost the same (when there is no packet loss) and thus, taking the mean has a limited impact on the overall energy consumption. Obviously, the relative error between ECOFEN and the homogeneous model will increase if this difference becomes larger. Hopefully, in real platforms this difference remains reasonable. Still, the heterogeneous energy model multiplies by three the number of links used in the SimGrid simulation. This is not negligible in terms of runtime on large-scale platforms.
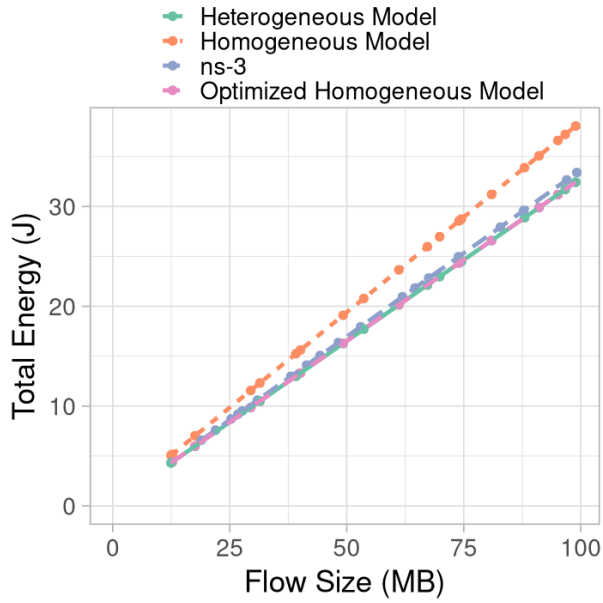
(a) 2 hosts 1 link platform power profile
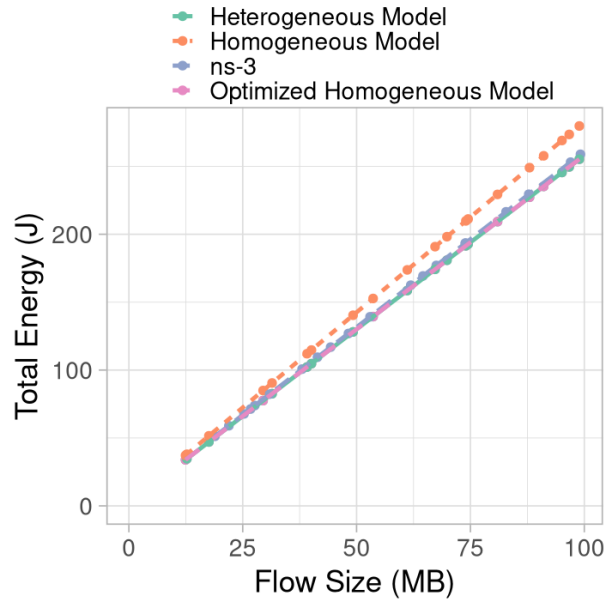
(b) 4 hosts 3 links platform power profile
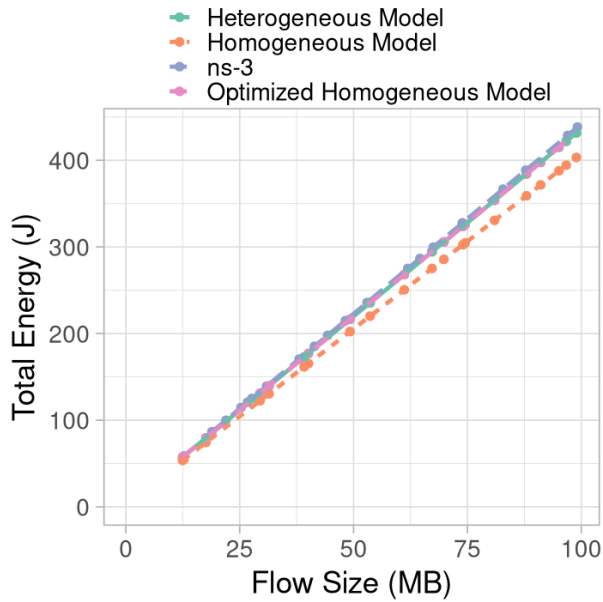
(c) Dogbone platform power profile

Figure 4.5: Microbenchmarks energy results on heterogeneous energy platforms.

This is why, by following the Occam's razor Principle, we decide to use the homogeneous energy model with the arithmetic mean instantiation in the remaining experiments.

4.4.2 Realistic use case

To evaluate the scalability of our approach, we propose to simulate a data center network which is a classical large-scale platform widely used in research [130, 76, 49]. Therefore, this final experiment has two main goals. First, it demonstrates that our optimized homogeneous energy model scales up in terms of execution time and memory usage. Secondly, it shows that this energy model is usable for real experiments and thus it is interesting for the scientific community. The data center platform is based on a classical three-tier architecture used in the Greencloud experiments [131]. It is composed of 8 core routers that provide access to the data center. The core nodes are linked to 16 aggregation switches by 10Gbps links. Note that each core nodes are linked to every aggregation switches. Then, these aggregation nodes are linked to 512 access switches by 1Gbps links that provide access to 1 536 servers by mean of 1Gbps links. Totally, this platform comprise 2 696 links. A scale down version of the platform is depicted on Figure 4.6. The latency of each link is fixed to 0.2ms which is the average latency measured between two Grid'5000 internal nodes. The platform is set with the energy consumption of 1Gbps/10Gbps links referenced in the literature [62, 127].

The simulation scenario is defined as follows. To avoid bias, we generate randomly between 10 to 400 external client requests. Each request is modeled using a 1MB TCP flows. The requests are generated and arrived simultaneously to the data center by the 8 core nodes. Next, the flows are spread randomly among the different aggregation switches and then reach the servers. Finally, the servers handle the requests and answer the clients. The experiments are launched on both simulators, SimGrid and ns-3 to compare their energy consumption estimation and also their performance in terms of execution time and memory usage. For each point of the simulation, we run 10 different experiments using different random seeds to better estimate the accuracy of our proposition.

The energy and scalability results are shown in Figure 4.7. The energy consumption results show that we have similar predictions on both simulators. The homogeneous model provides similar results to ns-3 with ECOFEN. Even in the worst cases, it still predicts energy values in the ECOFEN confidence interval. Using the same method provided by the Equation 4.7, we computed the ARE between SimGrid and ECOFEN. The homogeneous model exhibits an ARE lower than 4%, which is a reasonable accuracy regarding its level of granularity. Performance and scalability results are also shown on Figures 4.7b and 4.7c. The execution time is clearly
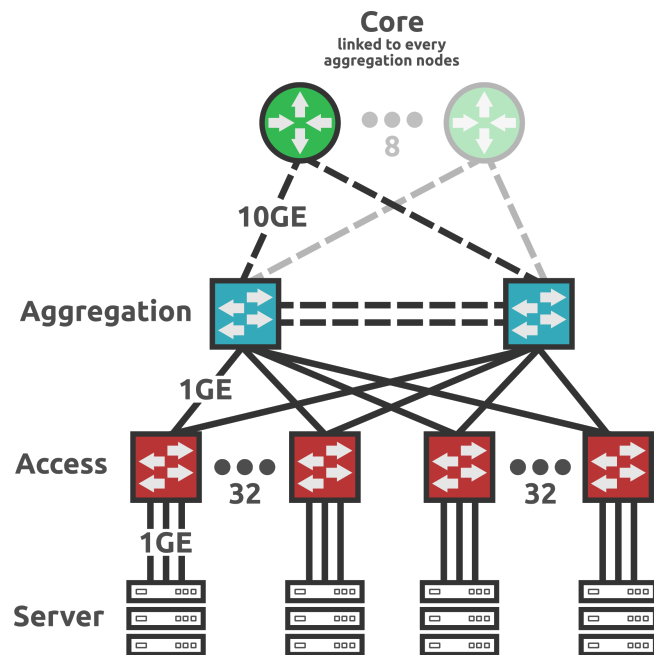
Figure 4.6: The three-tier data center platform used for the realistic use case simulations. The figure represents 1/8 of the original platform that contains 8 core routers and overall 192×8 servers.

higher on ns-3 than on SimGrid. In fact, for 258 requests, ns-3 takes more than 12 hours against only 6 minutes for 897 requests on SimGrid. This is why we stopped ns-3 experiments earlier than SimGrid ones. On this example, SimGrid is more than 120 times faster than ns-3 with ECOFEN. Similarly, the memory usage is also reduced: for 258 requests ns-3 requires 3GB of memory whereas SimGrid uses at most 169.08MB. It is logical since flow level simulators do not model every packets that are transferred from one host to another. Hence, their memory footprint and their computational overhead are drastically reduced. In this case, SimGrid uses 17 times less memory than ns-3.

(a) Overall data center power profile.

(b) Simulations execution time.
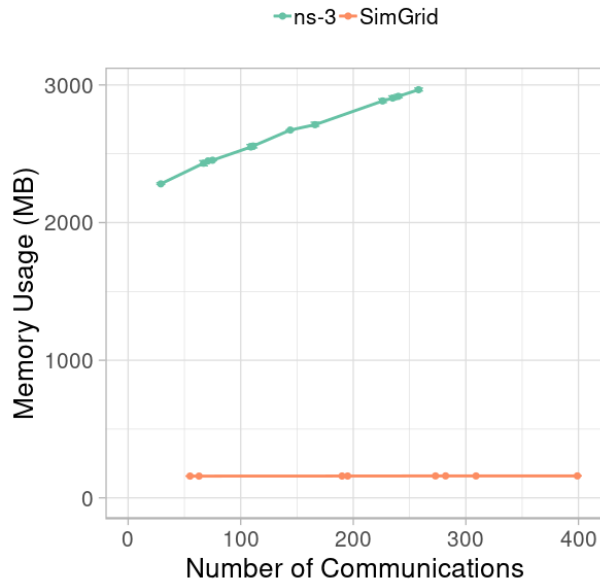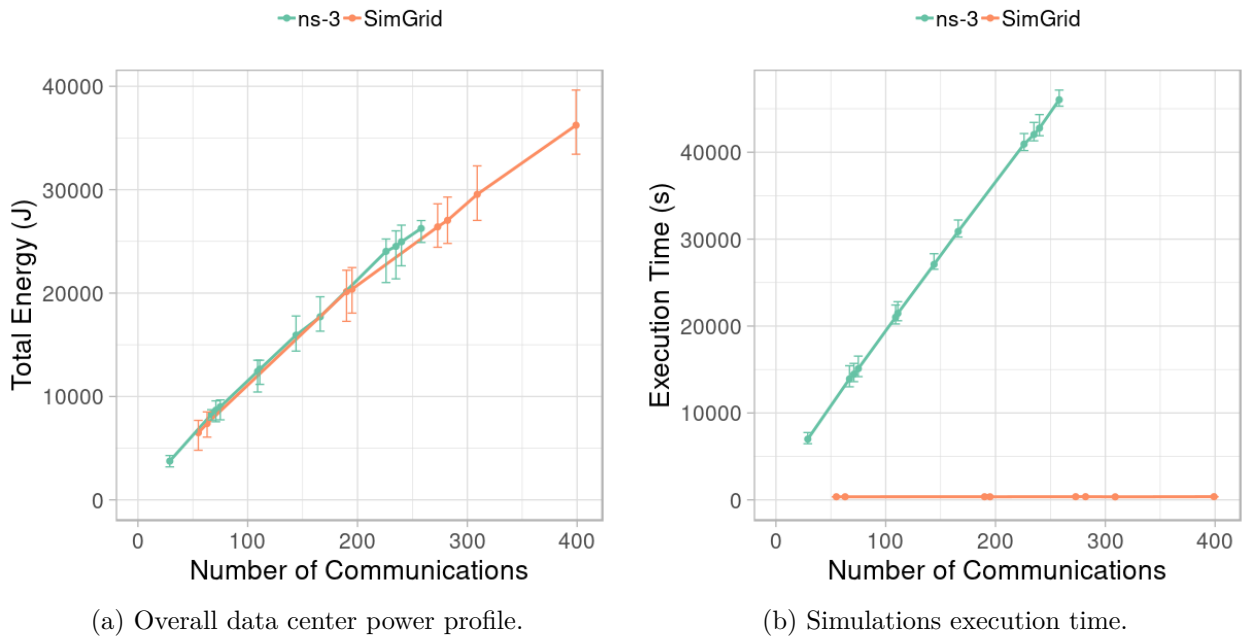
(c) Simulation memory usage.

Figure 4.7: Real use case energy and scalability simulations results.

4.5   Conclusion

In this chapter, we proposed a network energy model for flow based simulators. We showed that classical models have to be adapted for flow-based simulators due to their coarse-grained nature. To evaluate these energy models, we performed microbenchmark experiments. Their results show that FLSs can estimate the wired energy consumption with an accuracy close to PLSs. Additionally, we demonstrated that a fine grain modeling of platforms energy heterogeneity is not required to have accurate energy estimations. We also conduct data center network simulations which comprise 2 696 links. We were able to obtain accurate results with less than 4% of error and with a simulation duration 120 times smaller. This realistic use case highlighted how the model can be used by the scientific community on large-scale platforms.

However, even if our model provides accurate predictions for large-scale platforms and high bandwidth applications, it has some limitations. Indeed, it cannot be used for high precision measurements related to low-bandwidth applications. First, we assumed that every packets has a size equal to the MTU. Yet, in reality Ethernet packets size can range from 64 bytes up to the MTU. This can leads to inaccurate energy consumption predictions. Second, since our model is dedicated to FLS, it inherits from their accuracy and thus could not be used for high precision measurements such as bit flip effects or scenario with small variation of packet exchange (accurate protocol comparison, etc.). Still, the model can be used for wide variety of large-scale wired network platforms. Its implementation is open-source and available online [129].

This contribution allows to get closer to the end-to-end network energy consumption simulation. Since this model allows for wired energy consumption estimations, it is possible to use it for data center and ISP simulations. The logical next step towards our goal is to provide a scalable wireless performance model that could be used by devices located at the edge of the network.

PROPOSING AN EFFICIENT WI-FI SIMULATION MODEL

Contents

As discussed in Section 2.1.1, future IoT platforms will be composed of thousands of nodes communicating over the network. A large part of these nodes are using wireless technologies and Wi-Fi is one of the most used in today's terminals [6]. Moreover, its usage is expecting to become more and more intensive. In the last chapter, we proposed a wired network energy model suitable for large-scale experiments. The goal was to provide a model to study various parts of the end-to-end network such as Cloud, ISP and edge. In this chapter, we propose a scalable Wi-Fi performance model to study the wireless communications at the edge of the end-to-end network. As presented in Section 2.2, this work is a required step to study the Wi-Fi energy consumption.

This chapter is organized as follows. Section 5.1 presents the context and the problematic of the proposed Wi-Fi model. Section 5.2 introduces the core of the SimGrid bandwidth sharing

model and presents the Wi-Fi model along with its integration into SimGrid. Next, we evaluate the proposed model in Section 5.3 followed by a discussion in Section 5.4. Finally we will conclude in Section 5.5.

5.1   Context and Problem

Conducting efficient Wi-Fi network simulations is not a trivial task. We have seen in Section 2.3.4 that current Flow-Level Simulator (FLS) do not propose wireless performance models. Indeed, FLS target efficient wired network simulation and to the best of our knowledge, no Wi-Fi models have been proposed for FLS. Still, Wi-Fi simulations are possible using Packet-Level Simulator (PLS) such as ns-3, Komondor, OMNET++, etc. These Wi-Fi models are designed for accurate simulated time predictions. They implement many features of the Wi-Fi standards such as Ad-Hoc and Infrastructure models, accurate Distributed Coordination Function (DCF) function, rate adaptation, etc. Additionally, these Wi-Fi models are coupled to different physical models to account for wireless channel conditions as presented in Section 2.2.3. Consequently, the fine-grained nature of existing Wi-Fi models leads to scalability issues. Hence, current network simulators are not scalable enough to simulate Wi-Fi for large-scale platforms [85]. Most of them could not simulate more than a hundred of Wi-Fi nodes in a reasonable amount of time. As an example, in our end-to-end network study in Section 3 we faced scalability issues in ns-3 related to Wi-Fi simulations for a low-bandwidth application. Still, current use cases can involve high bandwidth scenarios as presented in Section 2.1.2. This pushes the scalability of current PLS to their limits even on simple network platforms.

In this chapter, to fill this gap between existing network simulators and real Wi-Fi network scenarios, we propose an efficient Wi-Fi model for large-scale platform simulations. This model aims at predicting the performance of a Wi-Fi cell in infrastructure mode. Thus, this model strives to account for the overall effects of the Distributed Coordination Function (DCF) allocation mechanism on the Stations (STAs) communication performance. However, to achieve efficient simulations, we neglect the following features of the actual Wi-Fi models:

- Ad-hoc mode

- Wireless channel models (path-loss, multi-path fading, etc.)

- Fine Distributed Coordination Function (DCF) behavior (RTS
  CTS mechanism)

- Modulations and signal properties (bandwidth, Coding Rate, etc.)

Similarly to the wired energy model detailed in Section 4, we propose to implement this Wi-Fi model into SimGrid. By doing so, we are extending the capabilities of SimGrid towards the end-to-end network energy framework as depicted on Figure 1.2.

5.2 An Efficient Flow-Level Wi-Fi Model

Our development of a coarse-grained Wi-Fi performance model suitable for flow-level network simulation is divided in two steps. First, a full understanding of the targeted simulation environment is required to clearly identify the scope of the final model. Then knowing this scope, it is possible to start defining the model. Thus, this section introduces the foundations of the Sim-Grid bandwidth sharing model. Next, the Wi-Fi bandwidth sharing model is proposed based on this preliminary study. Finally, an integration of the Wi-Fi model in the SimGrid bandwidth sharing model is given.

5.2.1 The core of SimGrid LMM Solver

The bandwidth allocation model is the heart of a flow-level network simulator. It ensures accurate time predictions. Since many predicted metrics are based on time predictions, the bandwidth allocation model of a FLS should be correctly designed. In SimGrid, this job is achieved by the Linear MaxMin (LMM) solver in charge of allocating the bandwidth for each communication that occur during the simulation [132]. The LMM solver is composed of an inequation system with variables and constraints. In this system, let C_r be the constraint associated with the inequation r, ρ_i be the variable i and finally $a_{r,i}$ the coefficients of the inequation r associated to the variable i. Then, the inequation system can be written as follows:

$$
\begin{cases}
\displaystyle\sum_{\substack{i \text{ using} \\ \text{link } 1}} a_{1,i} \times \rho_i & \leq C_1 \\
\qquad\vdots & \quad\vdots \\
\displaystyle\sum_{\substack{i \text{ using} \\ \text{link } r}} a_{r,i} \times \rho_i & \leq C_r \\
\qquad\vdots & \quad\vdots \\
\displaystyle\sum_{\substack{i \text{ using} \\ \text{link } m}} a_{m,i} \times \rho_i & \leq C_m
\end{cases}
$$

Thus, each link is represented by an inequation. The variables ρ_i represents the effective bandwidth (also called flows) to be allocated and the constraints C_r represents the capacity (bandwidth or allocable data rate) of each links. In addition, the coefficients $a_{r,i}$ can be applied to each variable. In classical network simulations, these coefficients can be used to model backward acknowledgments. Otherwise, they are equal to 1. Finally, each variable can be associated with a weight w_i. Weights prioritize flows among others and provide a way to model phenomenons such as RTT-Fairness. To solve the system, we have to maximize the bandwidth allocation starting from the most constrained inequation until we reach the less constrained one. Thus, to characterize how constrained an inequation is, we should introduce the load $\epsilon_r = \frac{C_r}{\sum_{i \text{ using } r} \frac{a_{r,i}}{w_i}}$ associated with the inequation n. Consequently, the next inequation to be solved is the one who minimizes ϵ_r. Finally, to solve a given inequation, each flow will have a resource allocation value of $\rho_i = \frac{\epsilon_r}{w_i}$. However, when a given ρ_i is already fixed, the constraint C_r of every inequation where ρ_i is involved should be updated with $C_r' = C_r - a_{r,i} \times \rho_i$. A complete overview of the LMM solver is proposed by Arnaud Legrand in [132]. Knowing these internals related to the SimGrid LMM solver, we are able to understand the conceptual choices behind the following Wi-Fi model.

5.2.2 The Flow-Level Wi-Fi Model

In a real wireless system, there is a lot of parameters that leverage communication performance. In fact, stations throughput highly depends on the communication channel. Several phenomenons such as signal-to-noise, interferences, path loss, multi-path fading have an impact on the communication performance. Our Wi-Fi model aims at neglecting all these physical phenomenons while keeping good prediction capabilities and achieving high performance simulations. The goal is to have a good representation of the Distributed Coordination Function (DCF) behavior on ideal channel conditions. To this end, out of any physical phenomenons, there is mainly two factors that leverage the Wi-Fi bandwidth allocation. The 802.11 specify that Wi-Fi stations transmit signal at a given rate r_i. This rate is defined by the physical layer capabilities of the stations and the access point. To communicate, each STA accesses the medium for a given duration called communication slot. Thus on a given time period called time slot, when a station is not using its communication slot, its resources should be redistributed among the other stations for the current time slot.

(a) Data sizes

(b) Time domain

(c) Durations
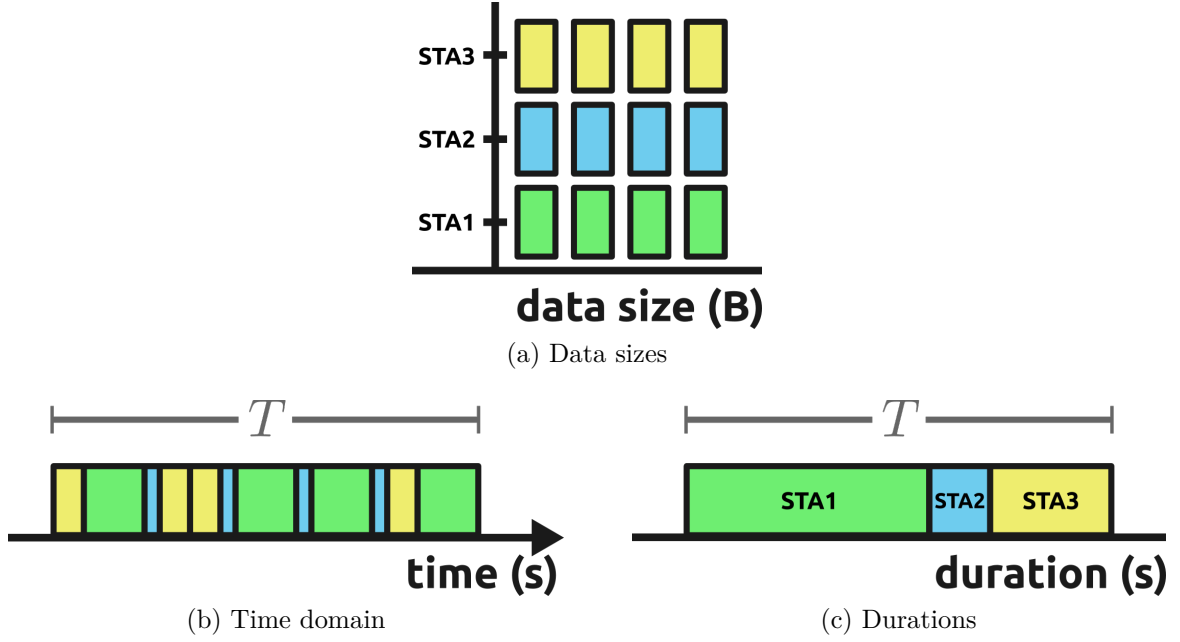
Figure 5.1: Example of a Wi-Fi communication time period T involving 3 STA communication slots. Each STA is communicating using its own data rate (r_i) according to its MCS index such that $r_1 = \frac{r_2}{2} = \frac{r_3}{4}$.

To build the model we consider a single Wi-Fi cell (in infrastructure mode) composed of n stations communicating at their own rate r_i and sending their own data d_i. Considering the coarse-grain serial communication for Wi-Fi presented in Figure 5.1. The overall cell capacity for a given time period T can be expressed as $C = \frac{\sum_{i=1}^{n} d_i}{T}$. However, as explained in Section 2.2.3, DCF tends to share the medium equally among the stations by forcing them to wait a random backoff time after each packet transmission. Since the backoff time is uniformly chosen [133] for all the stations, this leads to a fair sharing of the medium by the stations in terms of the amount of data d_i. Thus, we can have $\forall i, j \in (1, ..., n), d_i = d_j = d$ and thus $\forall i \in (1, ..., n), \rho_i = cste$. Therefore, system behave as if it had a virtual capacity C fairly shared between the n flows such that:

$$C = \sum_{i=0}^{n} \rho_i = \frac{n \times d}{T} = \frac{n \times d}{\sum_{i=0}^{n} \frac{d}{r_i}} = \frac{n}{\sum_{i=0}^{n} \frac{1}{r_i}} = \frac{1}{\frac{1}{n} \sum_{i=0}^{n} \frac{1}{r_i}} \tag{5.1}$$

This equation allows to easily embed the Wi-Fi sharing characteristics into the max-min sharing implemented in SimGrid for wired networks.

However, as said previously, depending on the simulation platform and scenario, many flows can be already fixed by other constraints. Lets suppose that the STA1's flow (noted ρ_f)

(a) Data size

(b) Time domain
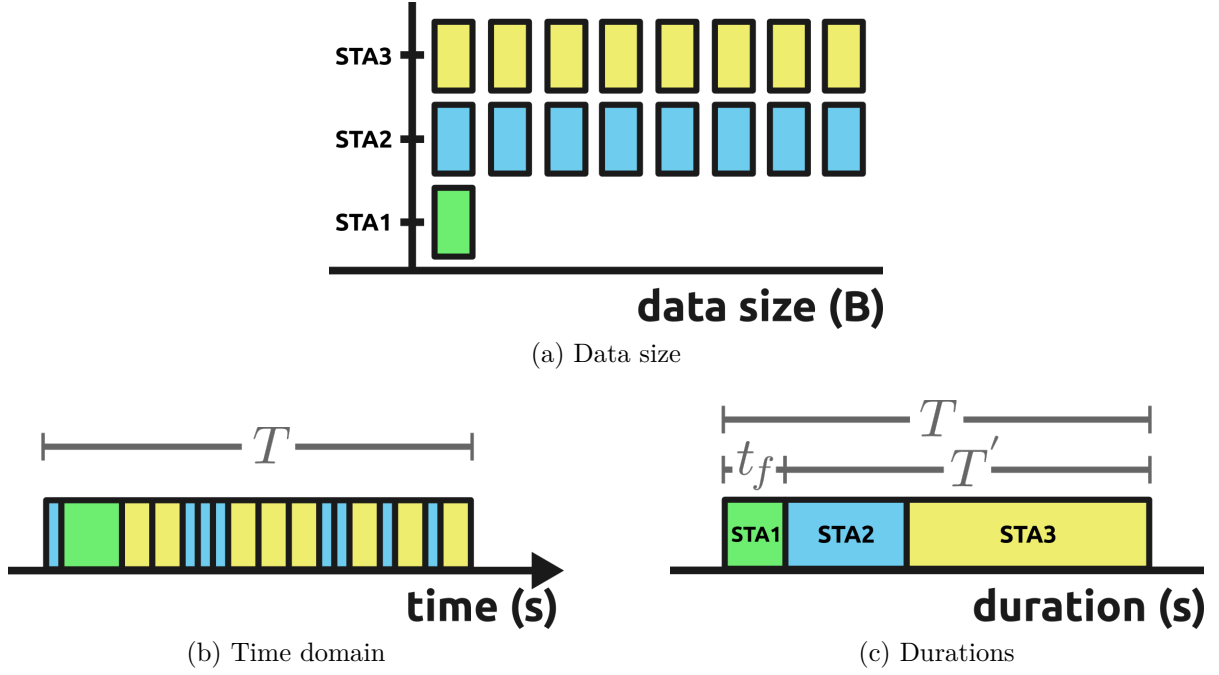
(c) Durations

Figure 5.2: Similar scenario than Figure 5.1 but with STA1's flow limited. t_f represents the total communication slot duration of the STA1 while T' represents the remaining duration on the time period T for the other STAs.

represented in Figure 5.1 is fixed such that $\rho_f < \frac{C}{n}$. Such a scenario can occurs if the receiver of the STA1's flow is not able to received as fast as STA1 can transmit. This scenario is represented on Figure 5.2. In that case, the non-fixed flows have a greater time to communicate and from their perspective, the antenna will behave as if it had a new capacity C' fairly shared among the remaining flows. Let us define d_f as the amount of data sent by the fixed flow such that $d_f \neq d$. Since $t_f = \frac{d_f}{r_f}$ and $\rho_f = \frac{d_f}{T}$ we have:

$$T = T' + t_f = T' + \frac{d_f}{r_f} = T' + T \times \frac{\rho_f}{r_f} \quad \text{hence} \quad T = \frac{T'}{1 - \frac{\rho_f}{r_f}}$$

Let us define d' as the amount of data that can be sent by the remaining STAs. Since we have:

$$C = \sum_{i=0}^{n} \rho_i = \rho_f + C' = \frac{d_f + (n-1)d'}{T}$$

From the perspective of the remaining flows, this capacity is such that:

$$C' = \sum_{\substack{i=0 \\ i \neq f}}^{n} \rho_i = \frac{(n-1)d'}{T} = \frac{(n-1)d'}{\frac{T'}{1-\frac{\rho_f}{r_f}}} = \frac{(n-1)d'}{T'} \times \left[1 - \frac{\rho_f}{r_f}\right] = \frac{1}{\frac{1}{n-1}\sum_{\substack{i=0 \\ i \neq f}}^{n} \frac{1}{r_i}} \times \left[1 - \frac{\rho_f}{r_f}\right]$$

In the general case, if we define I as the set of flows that remain to be fixed and F the set of fixed flows, we can generalize the expression above to any number of fixed flows with:

$$C' = \frac{1}{\frac{1}{|I|}\sum_{i \in I}\frac{1}{r_i}} \times \left[1 - \sum_{f \in F}\frac{\rho_f}{r_f}\right] \tag{5.2}$$

This last equation explains how the remaining virtual capacity of the antenna should be updated in the max-min algoritm of SimGrid whenever the bandwidth of some flow are fixed by contention on other resources. As expected, this model is easy to instantiate compared to packet-level Wi-Fi models. As expected, this model is easy to instantiate and expected to be very fast compared to packet-level Wi-Fi models. This model only requires few parameters which are the rate of each STA (r_i).

5.2.3 Integration Wi-Fi model into SimGrid

The last section defines the Wi-Fi model in terms of physical rate r_i and fixed flows. In this section we propose an integration of the model expressed by the Equation 5.2 into SimGrid. An elegant way to fulfill this goal is to reuse the actual SimGrid LMM solver detailed previously instead of creating a dedicated Wi-Fi LMM solver. This has several benefits. First, it minimizes the amount of additional source code introduced by the model. Second, it allows for easier multi-model simulations by combining wired and Wi-Fi models in the same simulation. Lets define ρ_i as the effective bandwidth allocated to the station i. According to the Equation 5.2, ρ_i can be expressed as:

$$\rho_i = C' \times \frac{1}{|I|} = \frac{1}{\sum_{i \in I}\frac{1}{r_i}} \times \left[1 - \sum_{f \in F}\frac{\rho_f}{r_f}\right]$$

Using the notations of Section 5.2.1 and define $\tilde{C}$ as the constraint used by the Wi-Fi cell inequation, we can impose that $\forall i \in (1, ..., n), w_i = 1, \tilde{C} = 1, a_i = \frac{1}{r_i}$. In the case of a fair share of this constraint, we will get:

$$\rho_i = \tilde{C} \times \frac{1}{\sum a_i} = \frac{1}{\sum_{i \in I}\frac{1}{r_i}}$$

Which is what we want. If there are fixed flows, meaning $F \neq \{\emptyset\}$, the new sharing is computed as follows:

$$
\begin{aligned}
\rho_i = \tilde{C} \times \frac{1}{\sum a_i} &= \left[1 - \sum_{f \in F} a_f \rho_f \right] \times \frac{1}{\sum_{i \in I} \frac{1}{r_i}} \\
&= \left[1 - \sum_{f \in F} \frac{\rho_f}{r_f} \right] \times \frac{1}{\sum_{i \in I} \frac{1}{r_i}} \\
&= C' \times \frac{1}{|I|}
\end{aligned}
$$

The modification to integrate the WiFi model with the Wired model is thus minimal as it only requires to change the instanciation (the a_i, w_i and the $\tilde{C}$) and the way the residual capacity is updated. Actually, in this model, the $\tilde{C}$ represents the fraction of channel time that can be shared between the flows. This implementation method suggests that Wi-Fi cells are modeled by a specific type of links called Wi-Fi links. A given link has an associated range of communication rates available to the STAs that corresponds to the differents MCS configurations. Then, each STA that are part of the same Wi-Fi cell shares the same link and uses one of the available rates to communicate. Thus, this integrated model requires to initialize two parameters namely the list of communication rates for the Wi-Fi cell and the actual communication rate of each STA.

5.3 Validation

In the previous section, we presented the coarse-grained Wi-Fi model implemented into Sim-Grid. In this Section, we propose a validation process to assess the model in terms of accuracy and scalability. First, we present the methodology used for the validation. Then, a calibration step is proposed prior to the experiments. Finally, we conduct an analysis of the results.

5.3.1 Methodology

As presented in Section 2.3.3, several packet-level network simulators are able to simulate Wi-Fi communication. In particular, ns-3 is well known for its 802.11 wireless models which have been used in numerous of scientific works [134, 85, 135]. To this end, we propose to use a similar approach than the one used in Chapter 4 by comparing our Wi-Fi performance model to the one of ns-3. In fact, this procedure is also used in similar work [85] related to Wi-Fi simulations.

To achieve this comparison, we develop two sets of experiments. The first set relies on an extension of SimGrid that implements the Wi-Fi model proposed in Section 5.2 by extending

the SimGrid LMM with the method presented in Section 5.2.3. The second set of experiments is based on ns-3 which implements the IEEE 802.11n standard. We choose to validate our model against this standard since it is one of the most used and experienced by current network simulators such as ns-3[136], OMNET++. Still, current IEEE 802.11ax remains compatible with the IEEE 802.11n which ensure its applicability in future networks. To validate our model, we compare the following metrics of each simulator:

- **Simulation Time** to compare time prediction on given data volumes to transmit (accuracy)

- **Throughput** to ensure bandwidth sharing consistency (accuracy)

- **Execution Time** to measure simulation efficiency (scalability)

- **Memory Usage** to compare both simulators memory footprint (scalability)

5.3.2   Experiment Settings

Prior to the validation process, it is important to carefully instantiate both network simulators. In particular, packet-level simulators comes with numerous parameters that are important to consider to have meaningful validation results. Lets first focus on the ns-3 simulator. At the physical and MAC layer, IEEE 802.11n can be configured in many different ways as stated in Section 2.2.3. Thus, what matters in our case is that:

Are we able to predict IEEE 802.11n performance for a given configuration ?

To answer this question, we choose a reasonable IEEE 802.11n configuration which is described in Table 5.1. First, we use the IEEE 802.11n at 5GHz with a channel bandwidth of 40MHz (with channel bonding) such as in [137] along with a guard interval of 800ns. Second, we choose a Modulation Coding Scheme of 3 which gives a theoretical throughput of 54Mbps. This implies the use of 1 spacial stream with a 16-QAM modulation at a coding rate of 1/2. In addition, to avoid any rate change during simulations, we disable the rate adaptation algorithm by using a constant rate model. For the same reasons, we also set the QoS to best effort. Then, to ensure stable communication throughputs, we enable RTS/CTS mechanism to mitigate the effects of hidden/exposed nodes. Finally, we keep the default aggregation mechanisms [138] which implies no MAC Service Data Unit (MSDU) aggregation (AMSDU) aggregation but provides MAC Protocol Data Unit (MPDU) aggregation (AMPDU) up to 65 535 bytes.

Name	Value
Carrier Frequency	5GHz
Bandwidth	40Mhz
Antennas Gain (Tx/Rx)	0dB
Transmit Power	0.1W (20dBm)
Rx Sensitivity	-101dBm
Guard Interval	800ns
RTS/CTS Threshold	100B
QoS	Best Effort
AMPDU	on
AMSDU	off
NSS	1
Coding Rate	1/2
Modulation	16-QAM
Rate Adaptation	ConstantRate

(a) Mac/Phy Layers Parameters

Name	Value
Propagation Delay	ConstantSpeed
Propagation Loss	FriisLoss
Error Rate Model	Nist [139]

(b) Channel Model Parameters

Table 5.1: Parameters of the ns-3 IEEE 802.11n performance model.

Regarding the channel configuration, we use two different models. First, the delay model uses the default constant speed model represented by the speed of light in vacuum. Then, we use a Friiz loss model to represent signal attenuation as explained in Section 2.2.3. The Friiz model is configured with a theoretical signal frequency of 5.18GHz $\approx$ 5GHz ($\lambda = 5.78e^{-2}m$) without antennas gain. Finally, since the IEEE 802.11n relies on OFDM we used the default Nist error rate model [139] which have has validated for OFDM communications to account for the signal resilience to the channel.

Given this IEEE 802.11n configuration for ns-3, the next step is to instantiate our SimGrid Wi-Fi model by determining the effective rate r_i of each station in the cell. Since we fixed the channel communication rate at 54MBps, we run a calibration experiment on ns-3 to determine the effective throughput of the actual configuration. The experiment consists in one station that communicates with an access point at a distance of 15m as depicted on Figure 5.3. Communications occur at the maximum achievable throughput with a TCP socket. This experiment gives us an effective throughput of 42.10Mbps. Thus, we use this value as the effective rate r_i for every station to instantiate our model. It is important to note that this rate will remain fixed during all the validation process since we are not using any rate adaptation algorithm.

To summarize, we configure the ns-3 simulator to reflect a IEEE 802.11n configuration with a single STA. Then we calibrate our SimGrid 802.11 model by measuring the station effective
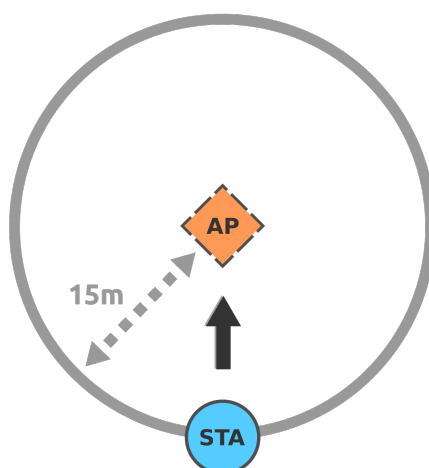
Figure 5.3: Calibration experimentation.

rate according to the rate delivered by ns-3 with the actual configuration. With this in mind, the next step is to pursue with the microbenchmarks simulations to examine the validity of our model.

5.3.3 Microbenchmarks description

The validation process consists in simulating several network scenarios (microbenchmarks) and compare both simulators outcomes on several metrics. Thus, we design 3 parametric platforms with different communication schemas (or flows). These platforms are depicted on Figure 5.4.

The first platform is noted P1. It is composed of several Wi-Fi stations communicating with an AP. Stations are distributed in circle, homogeneously (the angle formed with the AP and two consecutive stations is always the same) around the AP at a distance of 15 meters as used in the calibration process. This platform can be used with 1 to n stations as shown on Figure 5.4a. We use this platform as reference point. The idea is that, as long as simulations give valid results for this scenario, we can consider that our simulators are correctly instantiated. Thus, if for any reason, P2 or P3 experiments failed, it will not be the result of a wrong model configuration but rather due to a wrong scenario implementation or an invalidation of the model.

(a) Platform 1 (P1) with 4 STAs. (b) Platform 2 (P2) with 4 STAs. (c) Platform 3 (P3) with 3 STAs.
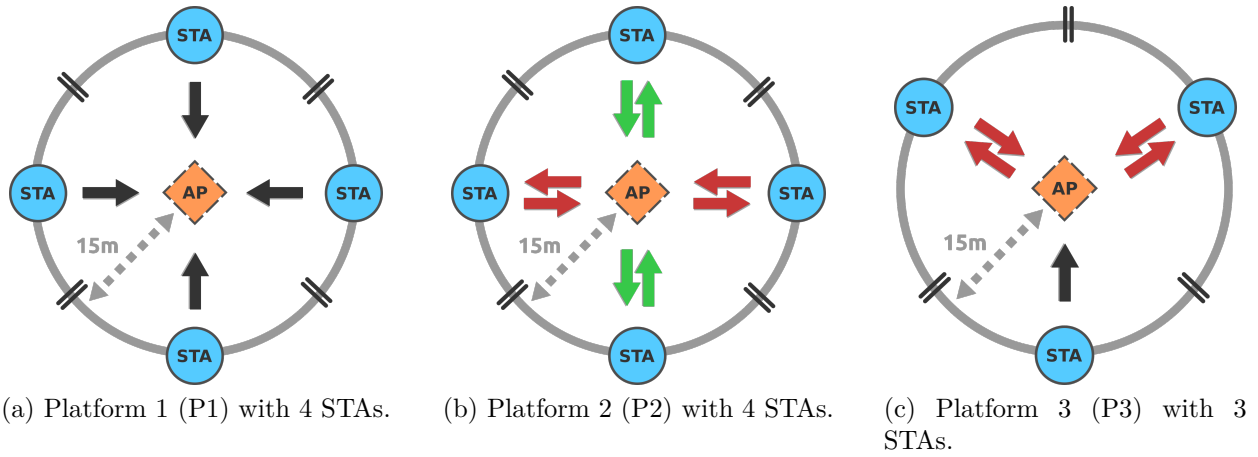
Figure 5.4: Microbenchmarks parametric platforms.

The second platform is noted P2. Spacial location and distribution of the stations and AP are similar to P1 but stations are now communicating by pairs. Thus P2 can be configured to be used with 2 to $2{\times}n$ stations as shown on Figure 5.4b. Communication between 2 stations occurs through the AP. These communications can be configured in two ways. Either unidirectional (STA A to STA B) or bidirectional (STA A to STA B and STA B to STA A). This scenario introduces more channel accesses and challenge the model bandwidth sharing capabilities.

Finally, the third platform P3 is an hybrid platform mixing P1 and P2. Similarly to P2, stations are communicating by pairs. But, for each pair of station, an additional station is added which communicate with the AP (as P1). Thus, P3 is using $3{\times}n$ stations as depicted on Figure 5.4c.

Totally, these three platforms lead to 5 differents configurations:

- P1 unidirectional (P1U)

- P2 unidirectional (P2U)

- P2 bidirectional (P2B)

- P3 unidirectional (P3U)

- P3 bidirectional (P3B)

For each of these 5 configurations simulations run with 4 different numbers of station. Each station is sending data ranging from 10Mb to 50Mb. In addition, each simulation on ns-3 is running with 20 random seeds (used in the loss models) to increase the results diversity. All

Name	Description
Platform	$\{P1U, P2U, P2B, P3U, P3B\}$
Distance AP/STA (radius)	15m
ns-3 seeds	$(s_1, ..., s_{20}) \in \mathbb{N}$
Data sent by stations	$(x_1, ..., x_{30}) \in [10Mb, 50Mb]$
Number of nodes	$n \in \mathbb{N}^+$ (platform dependent)
Total Number of simulation	12 000

Table 5.2: Microbenchmarks simulation parameters.

these simulation parameters are summarized in Table 5.2. By combining all the parameters, we reach a number of 12 000 simulations. Since these simulations involve 6 000 packet-level simulations with wireless communications, the microbenchmarks cannot be run on a single machine. Thus, all the simulations were run on Grid'5000 which is an experimental test-bed detailed in Section 2.1.1. The simulations are spread on 50 machines on the Rennes Grid'5000 site. The results gathered from this simulation campaign are analyzed and interpreted in the remaining of this section.

5.3.4   Accuracy Analysis on microbenchmarks

The results related to time and throughput predictions for the P1U platform are shown on Figure 5.5. The Figure 5.5a shows the simulated time predictions according to the amount of data sent by 1, 2, 5 and 15 stations. This figure shows a linear relation between the simulated time and the amount of data sent by each station. This linearity is possible thanks to the homogeneous positions of the stations around the AP. In fact, this station distribution allows for a fair sharing of the wireless channel since each station is in the same transmission conditions. In addition, the signal emitted by the stations reaches the access point with the same theoretical power. To measure the accuracy of the model, we used the metric presented in Section 4.4 called the Average Relative Error (ARE). The time prediction results show that our model does accurate predictions on the scenario with a single STA in the Wi-Fi cell. In this scenario, generated data shows an ARE of 0.18% for time prediction. This allows to conclude that our calibration process was well conducted. However, while the number of station is increasing, our model begins to diverge from the ns-3 model. The Table 5.3 presents the time and throughput ARE between SimGrid and ns-3. The results show that ARE reaches up to 26% with 15 stations. The Figure 5.5b shows the overall Wi-Fi cell applicative throughput predicted by each
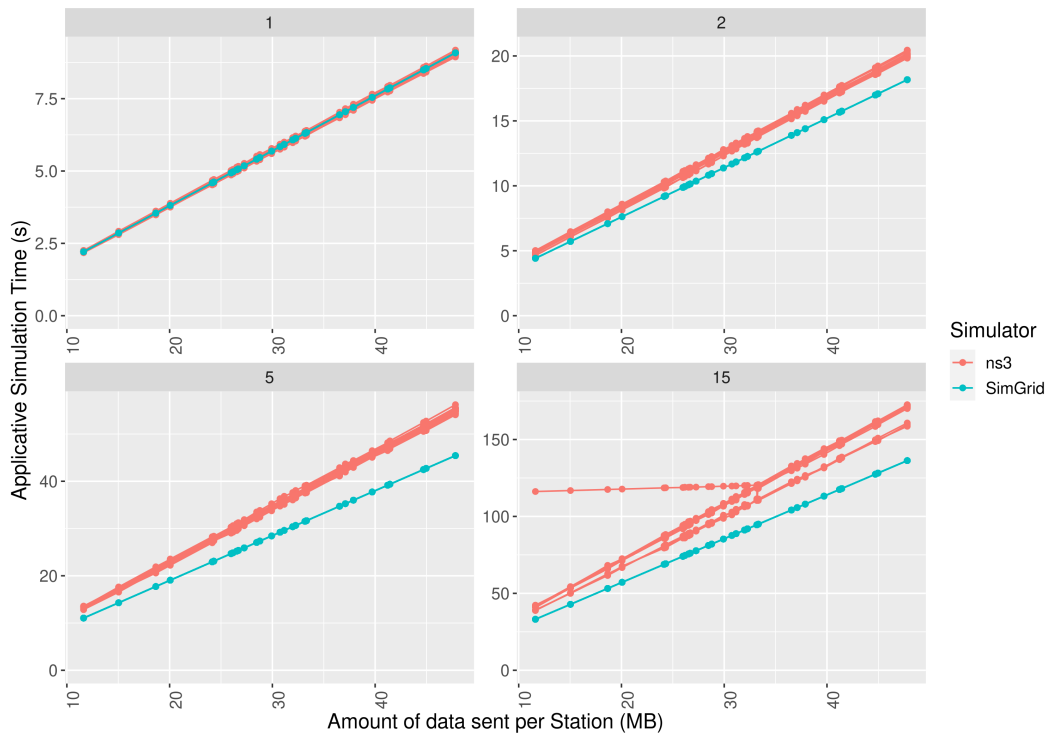
(a) Simulated time prediction of ns-3 and SimGrid for 4 different numbers of stations.
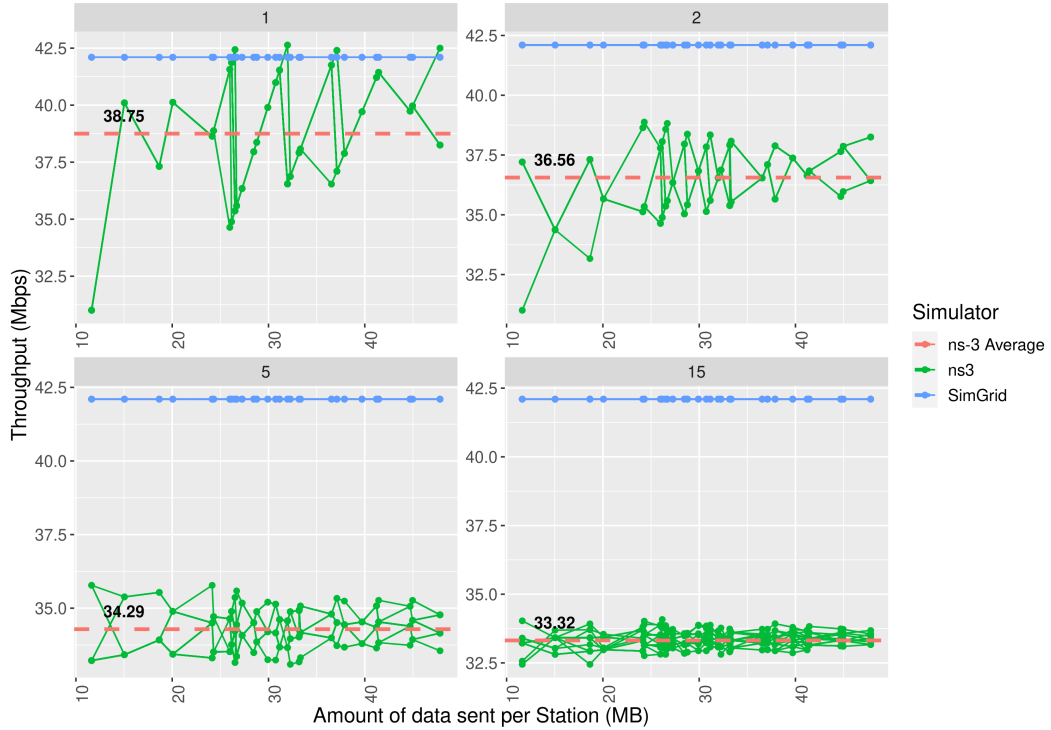
(b) Overall Wi-Fi cell throughput prediction of ns-3 and SimGrid for 4 different numbers of stations.

Figure 5.5: Microbenchmarks results for P1U platform.

P1U			P2U			P2B			P3U			P3B		
Number of Stations	T_{ARE}[a]	TH_{ARE}[b]	Number of Stations	T_{ARE}	TH_{ARE}	Number of Stations	T_{ARE}	TH_{ARE}	Number of Stations	T_{ARE}	TH_{ARE}	Number of Stations	T_{ARE}	TH_{ARE}
1	0.18%	8.64%	2	3.49%	8.07%	2	2.27%	4.61%	3	8.24%	11.65%	3	5.88%	7.84%
2	10.57%	15.16%	4	7.26%	9.98%	4	5.39%	6.67%	15	13.43%	14.12%	15	8.32%	8.76%
5	20.85%	22.79%	10	8.01%	9.01%	10	4.34%	4.85%	24	12.72%	13.50%	24	5.76%	8.10%
15	26.0%	26.36%	30	7.11%	7.47%	30	3.52%	4.34%	30	11.43%	13.12%	30	6.17%	8.34%

a. Time Average Relative Error
b. THroughput Average Relative Error

Table 5.3: Time prediction and throughput average relative error between SimGrid Wi-Fi model and ns-3 for each scenario.

simulator. These results confirmed the inaccuracies related to time predictions. Our model has difficulties in predicting the right throughput. Indeed we can see that the throughput is clearly overestimated by our model which leads to inaccurate simulated time predictions.
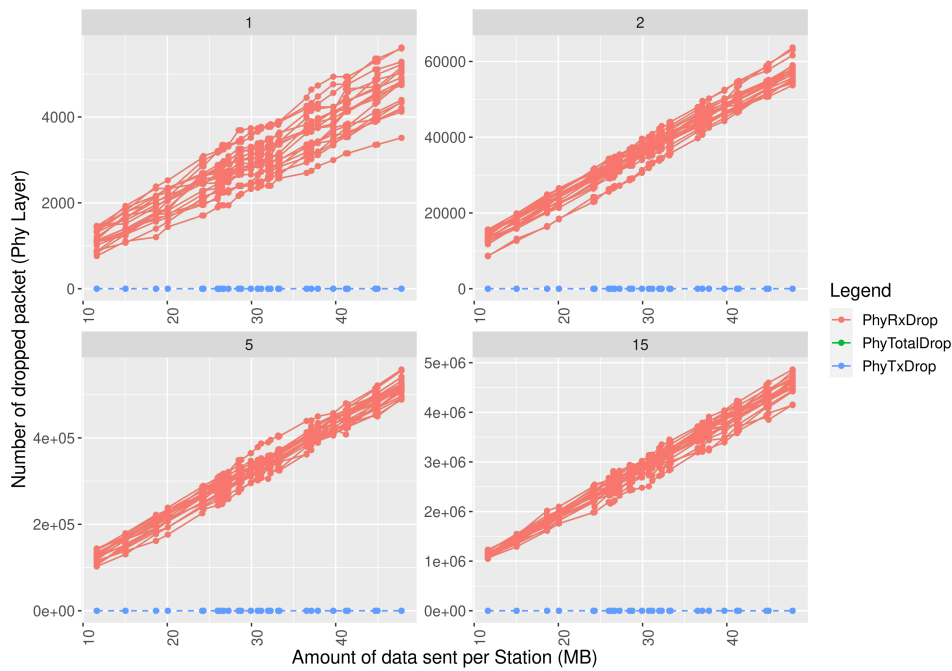
Figure 5.6: Analysis of packet dropped during P1U simulations.

The Table 5.3 shows similar trends for the other scenarios. Assuming that our model is valid, since we know that it is correctly instantiated it means that an additional model in ns-3 leverages the Wi-Fi throughput and impact our results. The data provided by Figure 5.6 shows the total amount of dropped packets in ns-3 during the Transmission (Tx) and the Reception (Rx) for the P1U scenario. This figure clearly shows that packets are essentially dropped on the

| Number of Stations | Interferences | | |
| | Enable | Disable | |
	Max PhyRxDrop	Max PhyRxDrop	Variation(%)
1	5618	5713	+1.691%
2	63683	53012	-16.746%
5	558128	466599	-16.399%
15	4862547	3167841	-34.852%

Table 5.4: Comparison of packet dropped during the P1U platform simulations in ns-3.

receiver side which leads to an over-approximation of the bandwidth by SimGrid. Since, our model accuracy seems to be inversely proportional to the number of stations, we can deduce that this issue is caused by interferences. Consequently, the interference model of ns-3 has a non negligible impact on the predicted bandwidth.

Thus, we run the exacts same scenarios while inhibiting the additive power interference model of ns-3 with the aims of isolating the effects of the Wi-Fi DCF. By doing so, the signal transmitted by the STA will not appear to interfere together on the receiver side which leads to an increase in the overall Wi-Fi cell bandwidth. Thus, we can expect to reach the bandwidth estimated by our model. These results for the P1U scenario are depicted on Figure 5.7. They show a significant improvement in the predictions accuracy. The P1U scenario has an ARE lower that 6% with 15 stations. The results for all the scenarios are provided in Table 5.5. All the microbenchmarks scenarios have improved in terms of time and throughput prediction accuracy. Nevertheless, it is difficult to fully characterize the evolution of the accuracy on very high bandwidth applications. Indeed, Figure 5.8 shows the evolution of the average relative error on both scenarios P1U and P3U. The results show that we cannot exhibit a clear tendency regarding the evolution of the accuracy on very large data transfers. But still, since both simulators have a linear time prediction, we can be confident that the accuracy will still be reasonable. In addition to predictions accuracy, the Figure 5.9 shows a significant reduction of packet drop for the P1U scenarios while using multiple stations. The Table 5.4 provides numerical results related to this reduction of packet drop. We can conclude that our model is able to predict the Wi-Fi bandwidth sharing mechanism with good accuracy while neglecting the interferences on the receiver side.
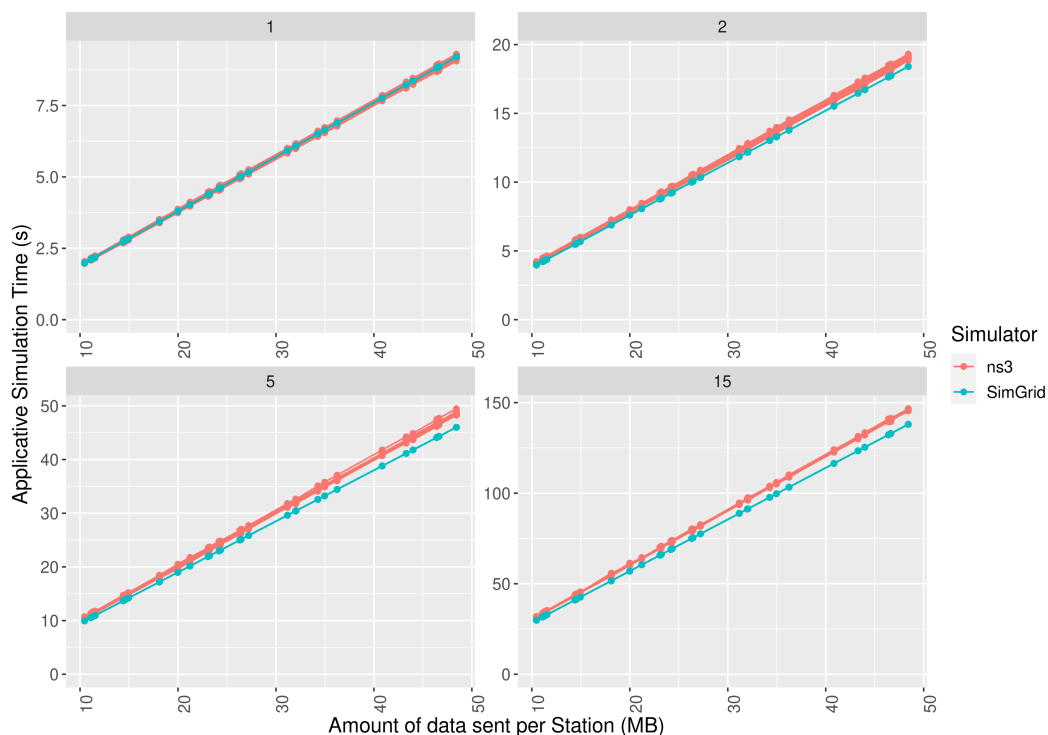
(a) Simulated time prediction of ns-3 and SimGrid for 4 different numbers of stations.
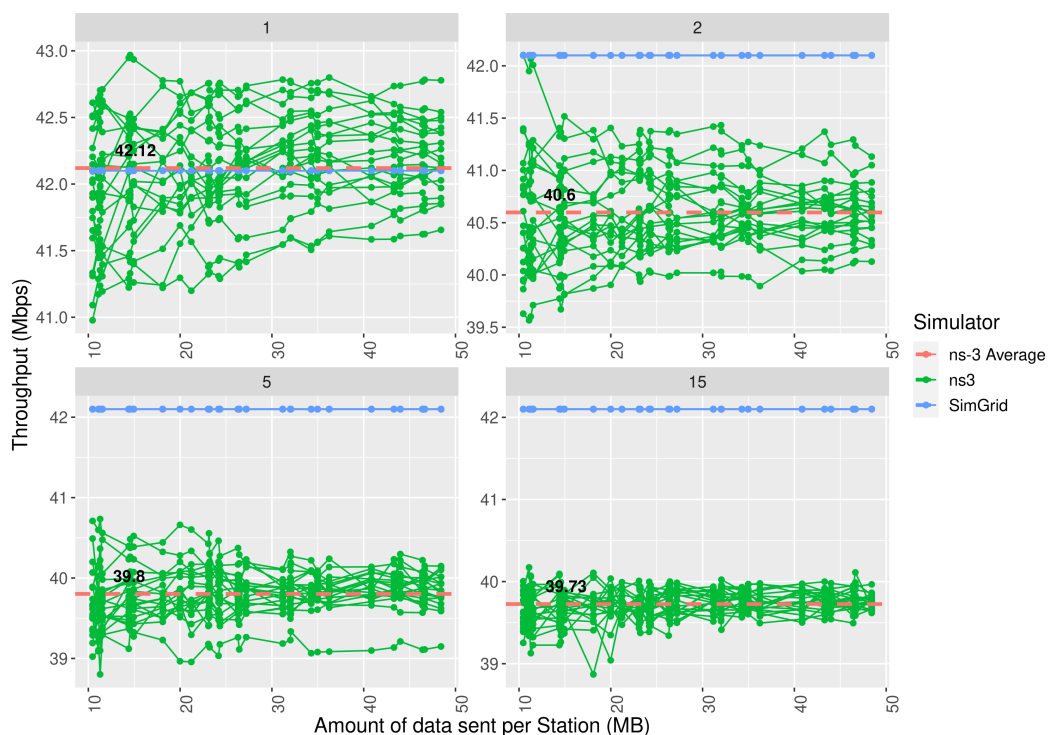
(b) Overall Wi-Fi cell throughput prediction of ns-3 and SimGrid for 4 different numbers of stations.

Figure 5.7: Microbenchmarks results for P1U platform without ns-3 additive power interference model.
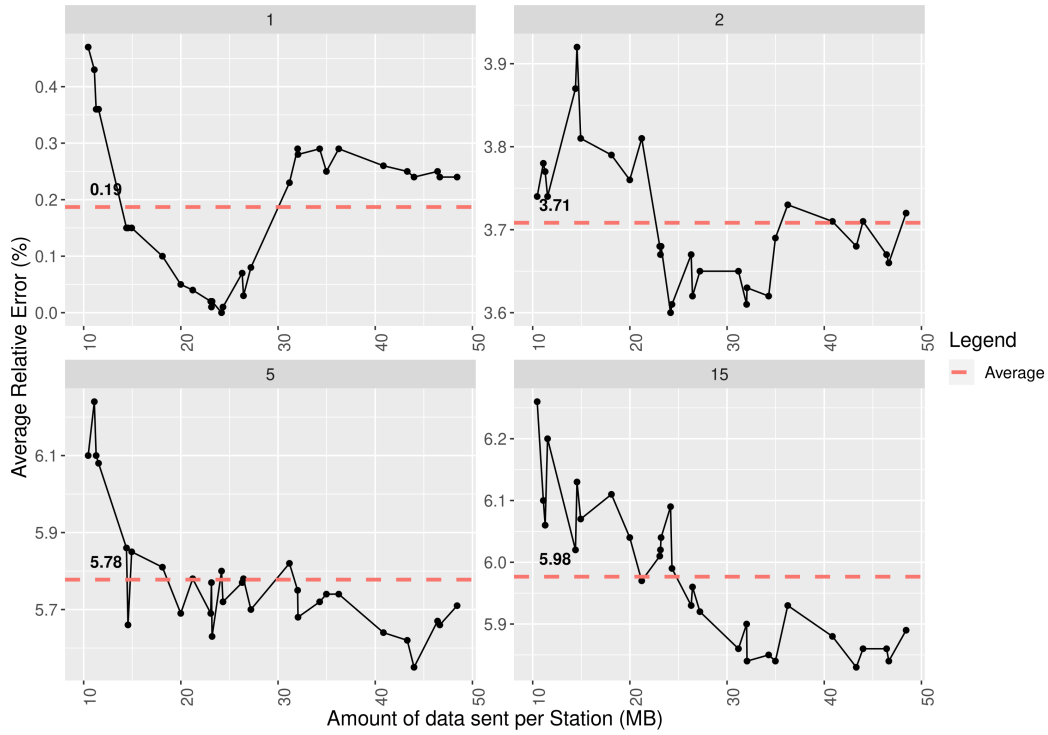
(a) Average relative error evolution on the P1U scenario.
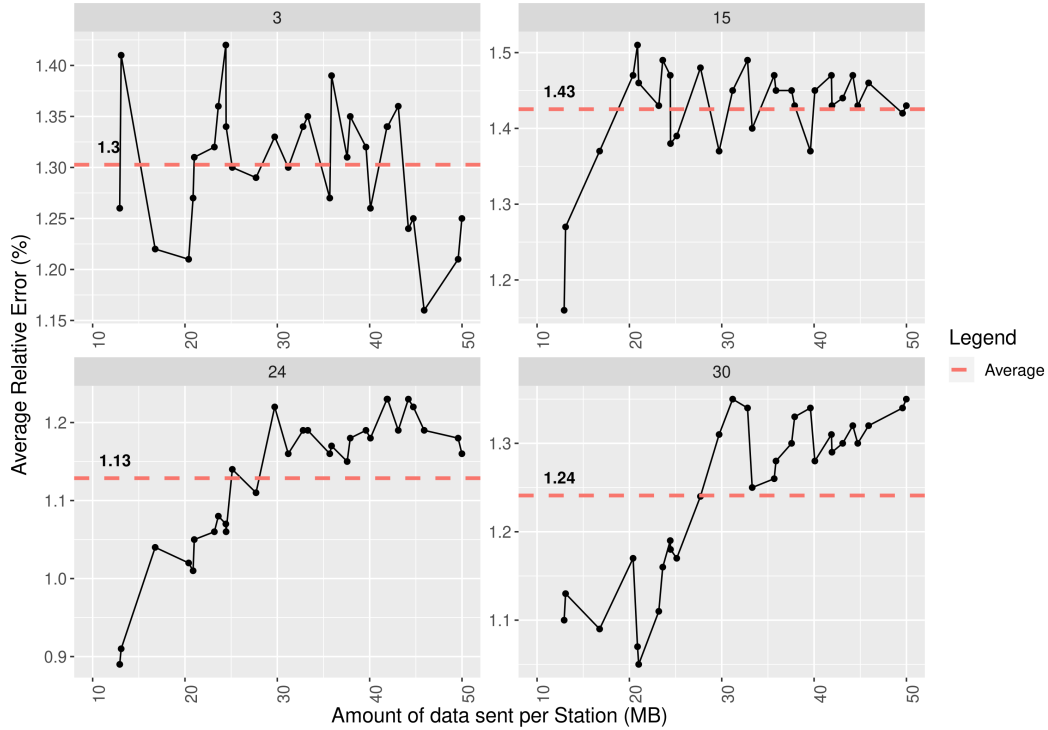
(b) Average relative error evolution on the P2U scenario.

Figure 5.8: Microbenchmarks average relative error evolution for P1U and P2U scenario without ns-3 additive power interference model.
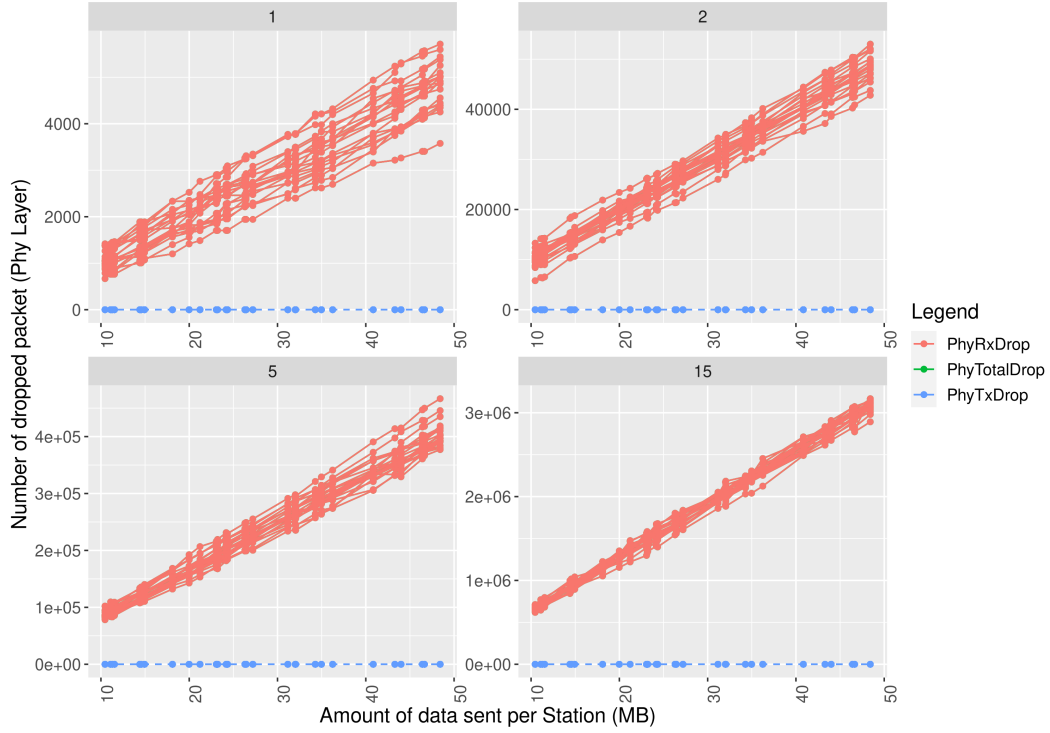
Figure 5.9: Analysis of packet dropped during P1U simulations without the ns-3 additive power interference model.

P1U			P2U			P2B			P3U			P3B		
Number of Stations	T_{ARE}[a]	TH_{ARE}[b]	Number of Stations	T_{ARE}	TH_{ARE}	Number of Stations	T_{ARE}	TH_{ARE}	Number of Stations	T_{ARE}	TH_{ARE}	Number of Stations	T_{ARE}	TH_{ARE}
1	0.19%	0.05%	2	0.14%	0.14%	2	1.46%	1.47%	3	1.30%	1.30%	3	1.00%	1.00%
2	3.71%	3.70%	4	0.46%	0.46%	4	1.41%	1.41%	15	1.43%	1.42%	15	2.75%	2.75%
5	5.78%	5.77%	10	0.19%	0.19%	10	3.85%	3.85%	24	1.13%	1.44%	24	3.04%	2.62%
15	5.98%	5.97%	30	0.72%	0.72%	30	4.55%	4.35%	30	1.24%	1.72%	30	2.41%	2.15%

a. Time Average Relative Error
b. THroughput Average Relative Error

Table 5.5: Time prediction and throughput average relative error between SimGrid Wi-Fi model and ns-3 for each scenario while disabling the ns-3 additive power interference model.
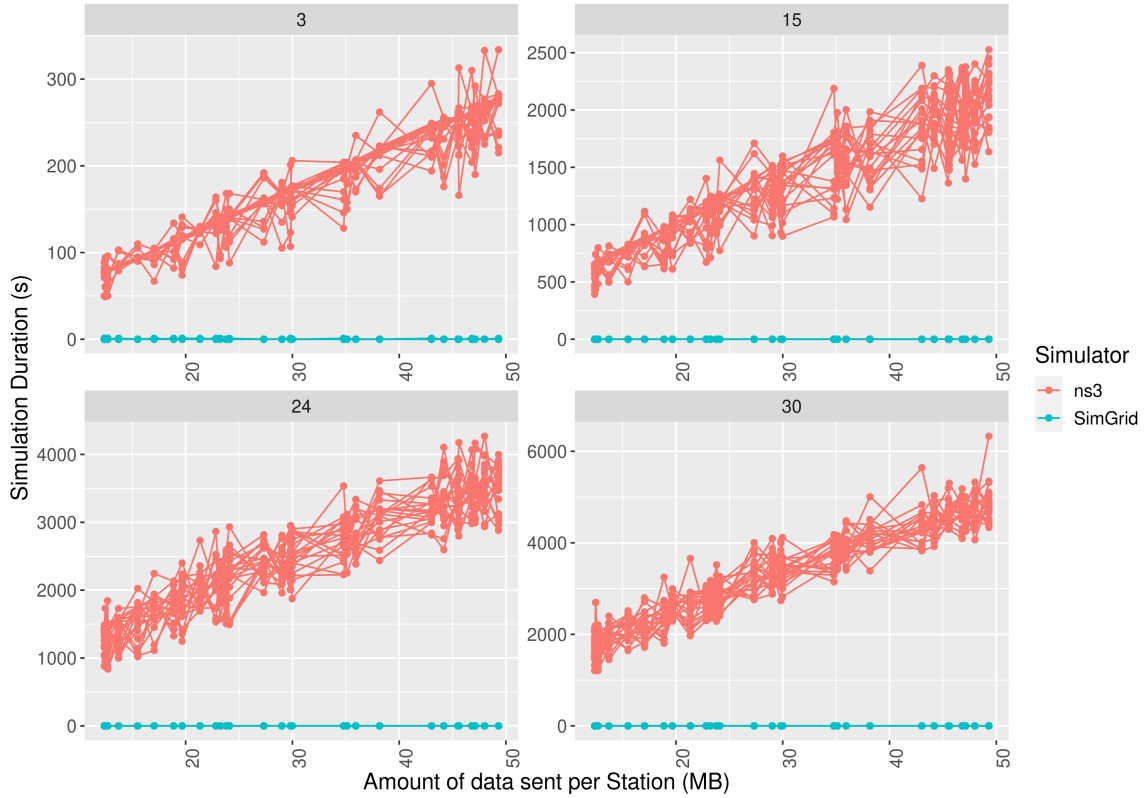
Figure 5.10: Analysis simulation duration of the P3B scenario.

5.3.5 Scalability Analysis on microbenchmarks

To demonstrate the scalability of our model, we keep track of the execution time and the peak memory usage of both simulators for every simulation. Figure 5.10 shows the simulation execution times according to the number of data sent by 1, 2, 5 and 15 stations on the P3B scenario. These results show that ns-3 has an execution time which evolves linearly with the amount of data sent over the network. As explained in Section 2.3.3, this phenomenon is specific to packet-levels simulators. Conversely, SimGrid execution time is totally independent of the amount of data sent during the simulation which makes it ideal for high bandwidth applications. In addition, we can see that SimGrid is outperforming ns-3 in terms of execution time. The figure shows that with 30 stations the maximum execution time of ns-3 corresponds to 1 hour and 45 minutes ($6\,328$s) compared to SimGrid which does not exceed 1s. Similarly, Figure 5.11 shows the peak memory usage of each simulation according to the number of data sent by each station. This figure reinforces the previous analysis on the execution time and shows that SimGrid has better memory footprint than ns-3. Even so, the memory usage of both simulator
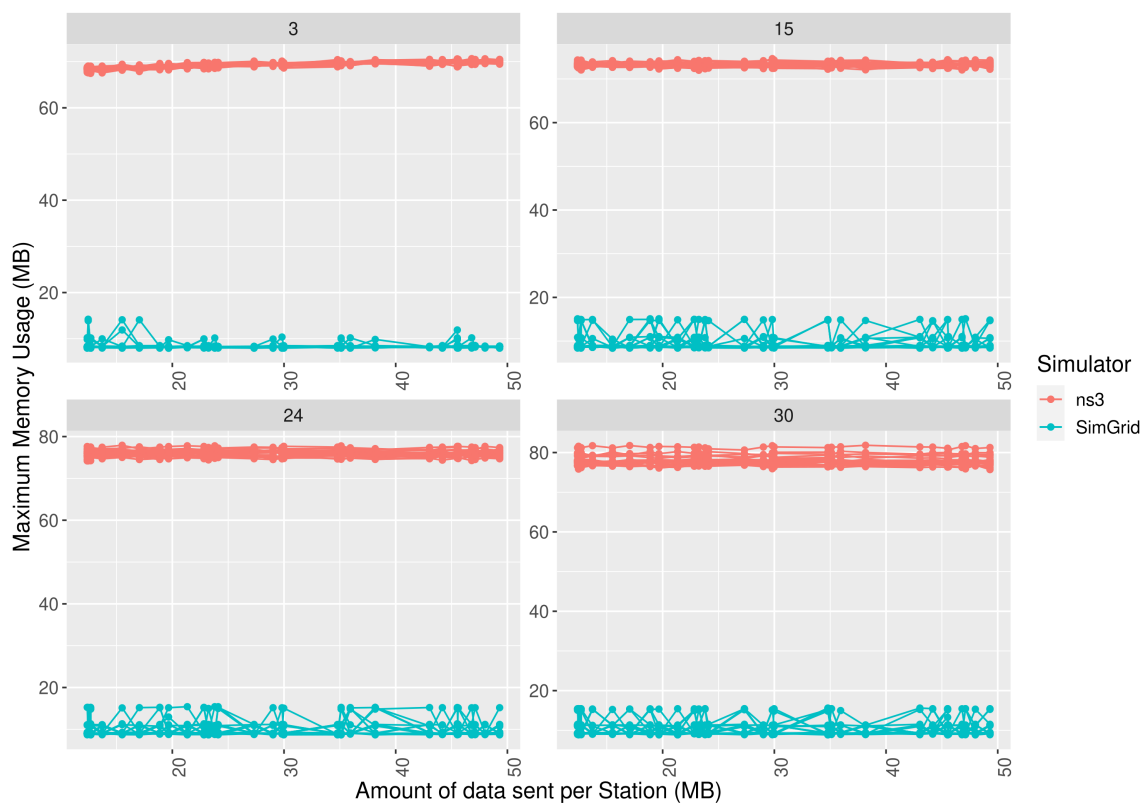
Figure 5.11: Analysis simulation peak memory usage of the P3B scenario.

is independent of the amount of data sent during the simulation. Indeed, since the network capacity is fixed by the initial platform, the amount of data exchange over the network cannot exceed this limit for a given simulation. Table 5.6 provides the scalability results for the P3B scenario. This table shows that the number of STA has a strong impact on the ns-3 execution time. Conversely, SimGrid execution time is almost unchanged. Thus in this scenario, we are able to estimate the performance of Wi-Fi for a given configuration with ARE of 3.04% in comparison with ns-3 in less than a second.

	ns-3		SimGrid	
Number of Stations	Execution Time	Peak Memory Usage	Execution Time	Peak Memory Usage
3	334s	70.65MB	<1s	14.22MB
15	2 527s	74.65MB	<1s	15.11MB
24	4 268s	77.92MB	<1s	15.41MB
30	6 328s	81.83MB	<1s	15.60MB

Table 5.6: Scalability analysis of the P3B scenario.

5.4   Discussion

The results for the microbenchmarks experiments show that our model is able to predict the Wi-Fi DCF bandwidth allocation mechanism on various scenarios without interferences. In addition, this model offers better scalability in terms of execution time and memory usage compared to packet-level Wi-Fi models. Still, several validation scenarios and research axes should be investigated:

Fixed flows scenarios: Our microbenchmarks did not validate scenario involving fixed flows. Indeed, fixed flows occur while a station communication rate is not limited by the Wi-Fi cell capacity itself (for example if the station communicates with a node located outside of the cell). In that case, the overall cell bandwidth allocated to the other stations should increase. Thus, experimentations with fixed flows should highlight this phenomenon.

Large scale experiment: Despite that our model were validated on several microbenchmarks scenarios, it could be interesting to validate it against more realistic scenarios. This could involves multiple Wi-Fi cells along with wired communications and realistic network traffic. In this way, it will reinforce the confidence in the model and provide better insight on its scalability capabilities.

Rate adaptation: In a real Wi-Fi deployment, the Access Point could handle stations with heterogeneous communication rates. Moreover, common operating systems such as Linux offer Wi-Fi rate adaptation algorithms such as Minstrel to optimize the communication performance. These heterogeneous rates and rate adaptation scenarios were not considered in our experiments since we used a fixed IEEE 802.11n configuration (using a given MCS). Thus, extensive validation experiments should be conducted in this regards.

Interference model: These microbenchmarks experiments pinpoint that an additional coarse-grained interference model is needed to predict the Wi-Fi performance correctly. In reality all the Wi-Fi cells are subject to interferences and thus this point should be considered

in the future.

5.5 Conclusion

Today's Wi-Fi simulation models lack for scalability. Thus they cannot be used for large-scale end-to-end network energy studies. As a response, we proposed a scalable Wi-Fi bandwidth sharing model to approximate the Wi-Fi DCF bandwidth sharing mechanism in infrastructure mode. Next, we proposed an implementation of the model into the SimGrid LMM solver. Then we validated the model against microbenchmarks scenarios. The results show good predictions capabilities with a average relative error less than 6% for all the considered scenarios. Additionally, our model offers great scalability features and outperforms the performance of the common packet-level simulator ns-3 in terms of execution time and memory usage. As discussed previously, an additional validation campaign should be conducted regarding rate adaptation, large-scale experiments, fixed flows and interferences.

This Wi-Fi performance model was proposed with the aim of studying the end-to-end networks energy consumption. This model can be used as a wireless communication model for the edge part of the network as presented on Figure 3.1b but it is currently limited to ideal channel with no interferences. To estimate the energy consumed by the Wi-Fi devices, an additional energy model is required. By combining our performance and Wi-Fi energy model, we should be able to predict the energy consumed by Wi-Fi communications as expressed in the Equation 2.1. This perspective opens up new avenues for studying IoT scenarios on very large-scale.

CONCLUSION AND PERSPECTIVES

Contents

Regarding large-scale networks energy consumption studies of Fog infrastructures, simulators offer a great alternative to real experimentations. Simulation-based studies are common in the literature. However, current simulators and models were not scalable and versatile enough for large-scale studies. Our solution is to propose scalable models in a single simulator to make these network energy studies possible at large-scale, from the IoT devices to the Cloud servers hosting their applications.

6.1 Conclusion

Fog computing and IoT paradigms lead to an increase in complexity of network platforms. Indeed, modern network platforms are composed of many heterogeneous nodes located at the edge of the network with a variety of network footprints and nodes located at the core. Moreover, the network traffic is increasing with the development of high bandwidth applications such as video streaming. This evolution of the Internet usage impacts the energy consumption of networks. These additional connected objects and Fog processing nodes are new ICT devices energy consumers. With the aim of reducing their energy consumption and their impact on the overall network, scientists have to study these large-scale platforms. However conducting real experiments remains difficult since platforms are very large. The approach chosen in this thesis was to study large-scale platforms by mean of simulations which offer several benefits such as

time, money saving and experiments reproducibility.

In the first place, we proposed an end-to-end energy study. Besides providing an interesting end-to-end energy analysis, this study revealed multiple flaws raised by current experimentations tools. First, none of them is able to cover the range of platforms deployed in today's Internet such as IoT, Fog, ISP networks and Cloud. Hence, the experimentation protocol has to be divided in several parts (one for each platform). Then, each of these platforms has to be studied separately using differents experimentation tools. Morever, in real networks these platforms are interacting together. This leads to difficulties studying the interactions between the platforms and having continuous communication flows between the IoT platforms up to the Cloud servers on a common ISP network. Consequently, network communications have to be piped between each part of the experimentation protocol to obtain consistent results. Regarding network simulations, our study used a common PLS called ns-3 which suffers from scalability issues. Thus we were not able to pursue our end-to-end study on large-scale IoT scenarios.

To solve all these issues, we proposed to extend a FLS called SimGrid by providing scalable network models for energy and wireless communication studies. Our first model was a wired network energy model dedicated to wired large-scale network platforms. This model can be instantiated with classical packet-level model parameters such as the energy consumption values per processed byte and per packet. We validated this model by mean of simulations. We compared our model to the results from the ECOFEN module of ns-3. These results showed that our model has accurate predictions along with high scalability capabilities. With this model we are now able to study the energy consumption of large-scale network platforms which involve wired communications such as ISP networks and data centers in a single simulation tool.

To extend the studies up to the edge of the network, we proposed a coarse-grained Wi-Fi performance model for SimGrid. This model strives to estimate the performance of Wi-Fi in Infrastructure mode where stations can communicate through an AP. Common packet-level Wi-Fi models require to setup many parameters while our model can be configured by specifying only the communication rate of the STA (known as MCS). This reduction of complexity has led to a highly scalable Wi-Fi model. We also proposed a first validation step by mean of network simulations. We compared the results given by the 802.11n model of ns-3 to our model. These first results showed accurate performance predictions under ideal wireless communication channel conditions (without interferences). Still, our model does not take into account the rate adaptation feature of common Wi-Fi devices. In addition, our study was limited to microbenchmarks and requires more intensive validation scenarios. Yet, it is a first step toward efficient Wi-Fi simulations and we are now able to simulate Wi-Fi cells under ideal conditions

at large-scale and in an efficient manner.

These contributions allow for larger network energy studies and we are now close to propose a complete end-to-end network energy simulation framework. Cloud and ISP network energy consumption can be studied using a single network simulator. With an additional Wi-Fi model validation regarding fixed-flows and the development of an interference model we will be able to study the entire network from the IoT, the Fog, the ISP network and the Cloud. Still, our work also leads to many future work in the domain.

6.2   Future Directions

6.2.1   Wi-Fi Enhancement

Our current Wi-Fi model faces limitations. The experiments conducted in our study showed that the Wi-Fi model is inaccurate on scenarios with interferences. To this regard, proposing an interference model for Wi-Fi in Infrastructure mode is an interesting future direction to consider. Indeed we have seen that the accuracy of our model is correlated with the number of stations that communicate through the wireless channel. Thus, grounding the interference model on the number of stations would be a good starting point toward accurate performance predictions.

In addition, Wi-Fi energy consumption prediction is an interesting feature to introduce. Indeed, currently our simulation framework proposes CPU and wired network energy model but lack of a wireless energy model. Similarly to the wired network energy model, this new model would be based on packet-level parameters which involve byte and packet energy consumption values. Then, based on performance predictions of our Wi-Fi model it would be possible to predict the energy consumption of Wi-Fi communications and thus provide a complete end-to-end network energy framework.

6.2.2   Support for mobility

In today's networks it is common that wireless technologies involve mobility. In typical PLS, this feature is insured by the propagation loss model and rerouting mechanisms. Depending on the distance from the STA to the AP, the propagation loss model will modulate the strengh of the signal received on the receiver side accordingly. However, due to their coarse-grained nature FLSs do not provide such a model. One way to approach flow-level mobility would be to leverage the STA communication rate according to its distance from the AP. Thus, the farest

Table 6.1: Exemple of studies that can benefits from an end-to-end simulation framework.

Document	Domain	Exp. Tool[a]	Wireless Tech.[b]	Platform	Metrics[c]
S. Tahir et al. [141]	eHealth	iFogSim	Bluetooth	End-to-End	Energy
A. Mebrek et al. [142]	Fog	Custom[d]	—	End-to-End	Energy,QoS
G. Li et al. [143]	Fog	Matlab	—	End-to-End	Energy,Delay
Z. Chang et al. [144]	Fog	Custom	—	IoT,Fog	Energy
F. Jalali et al. [5]	Fog	Custom	Wi-Fi,Ethernet,4G	End-to-End	Energy
H. O. Hassan et al. [145]	Fog	Custom	—	End-to-End	Energy,Delay,QoS
R. O. Aburukba et al. [146]	Fog	Custom	—	End-to-End	Delay
S. Sarkar et al. [147]	Fog	Real Deployment	Wi-Fi,Zigbee	End-to-End	Delay
M. M. Mahmoud [148]	eHealth	iFogSim	—	End-to-End	Energy
A. Toor [149]	Fog	iFogSim	—	End-to-End	Energy,Delay

a. Experimental Tool used in the work
b. Wireless Technologies used on the edge part
c. Metrics studied by the work
d. Use a custom network simulator based on a numerical model

the STA is from the AP, the lowest would be its communication rate. But, to determine this attenuation model, experimentations should be conducted and a new mobility model should be designed.

6.2.3 Support for other wireless technologies

Despite being used by a lot of ICT devices, Wi-Fi is not the single technology used in IoT and Fog infrastructures. Since connected objects are meant to be connected for a long period, many wireless technologies have been developed targeting low power energy consumption. Thus, proposing additional wireless communication models could improve the range of applications of our simulation framework. Table 6.1 is presenting studies related to Fog infrastructures. This table reveals that other technologies such as Bluetooth, Zigbee, 4G or LoRa [140] could be used for end-to-end network studies. Still many other wireless technologies are suitable for the IoT and should be considered as future work for our simulation framework.

6.2.4 Future research impact

In the literature many works can benefit from an end-to-end simulation framework. Table 6.1 revealed interesting characteristics about them. First, most of the presented work is related to end-to-end energy studies and involves IoT, Fog, ISP and Cloud platforms. However, the majority of them based their experimentations on a custom numerical model which is often

not subject to validation. Indeed, wireless communications are usually modeled by a delay function while the wired energy consumption is often neglected. Thus, providing an end-to-end simulation framework would speed up their experimental process while providing validated models and increasing their simulations flexibility. Still, S. Tahir et al. [141] used iFogSim as an experimental tool in an end-to-end scenario that involved Bluetooth communications. Thus, the accuracy of such study can be greatly improved with an end-to-end simulation framework that provides the proper models. S. Sarkar et al. [147] proposed a real experimentation deployment with accurate measurements to study Fog infrastructure latencies. A similar approach can be used to instantiate simulations. Using an end-to-end network simulation framework to study the energy and the performance could greatly benefits to the research community. Ideally, on the long term such a framework should meet the following requirements:

- Accuracy (containing validated performance and energy models)

- Scalability (handling large-scale platforms and high bandwidth scenarios in a reasonable amount of time)

- Versatility (suitable for IoT, ISP networks and Cloud)

- Adaptability (including Ethernet, Wi-Fi, 802.15.4, LoRa, etc. with easy means to extend for future technologies)

- Simplicity (models should be simple to use and to instantiate)

BIBLIOGRAPHY

[1] S. Diouani and H. Medromi, "How energy consumption in the cloud data center is calculated", *in 2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, Agadir, Morocco: IEEE, Jul. 2019, pp. 1–10, ISBN: 978-1-72810-827-8. DOI: `10.1109/ICCSRE.2019.8807458`. [Online]. Available: `https://ieeexplore.ieee.org/document/8807458/`.

[2] K. Georgiou, S. Xavier-de-Souza, and K. Eder, "The IoT Energy Challenge: A Software Perspective", *IEEE Embedded Systems Letters*, vol. 10, *3*, pp. 53–56, Sep. 2018, ISSN: 1943-0663, 1943-0671. DOI: `10.1109/LES.2017.2741419`. [Online]. Available: `https://ieeexplore.ieee.org/document/8012513/`.

[3] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-LAB: A large scale open experimental IoT testbed", *in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy: IEEE, Dec. 2015, pp. 459–464, ISBN: 978-1-5090-0366-2. DOI: `10.1109/WF-IoT.2015.7389098`. [Online]. Available: `http://ieeexplore.ieee.org/document/7389098/`.

[4] Cisco, *Cisco Annual Internet Report*, 2020.

[5] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing", *IEEE Journal on Selected Areas in Communications*, vol. 34, *5*, pp. 1728–1739, May 2016, ISSN: 0733-8716. DOI: `10.1109/JSAC.2016.2545559`. [Online]. Available: `http://ieeexplore.ieee.org/document/7439752/`.

[6] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022", Tech. Rep., 2019, p. 33.

[7] T. Chen, X. Gao, and G. Chen, "The features, hardware, and architectures of data center networks: A survey", *Journal of Parallel and Distributed Computing*, vol. 96, pp. 45–74, Oct. 2016, ISSN: 07437315. DOI: `10.1016/j.jpdc.2016.05.009`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0743731516300399`.

[8] *Grid'5000*. [Online]. Available: `https://www.grid5000.fr/`.

[9] L. Ramakrishnan, A. Liu, P. T. Zbiegel, S. Campbell, R. Bradshaw, R. S. Canon, S. Coghlan, I. Sakrejda, N. Desai, and T. Declerck, "Magellan: experiences from a science cloud", *in Proceedings of the 2nd international workshop on Scientific cloud computing - ScienceCloud '11*, San Jose, California, USA: ACM Press, 2011, p. 49, ISBN: 978-1-4503-0699-7. DOI: `10.1145/1996109.1996119`. [Online]. Available: `http://portal.acm.org/citation.cfm?doid=1996109.1996119`.

[10] R. Ricci and E. Eide, "Scientific Infrastructure for Advancing Cloud Architectures and Applications", vol. 39, *6*, p. 3, 2014.

[11] M. Winther, "WHITE PAPER Tier 1 ISPs: What They Are and Why They Are Important", p. 13,

[12] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys & Tutorials*, vol. 17, *4*, pp. 2347–2376, 2015, ISSN: 1553-877X. DOI: `10.1109/COMST.2015.2444095`. [Online]. Available: `http://ieeexplore.ieee.org/document/7123563/`.

[13] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges", *IEEE Communications Surveys & Tutorials*, vol. 20, *3*, pp. 1826–1857, 2018, ISSN: 1553-877X, 2373-745X. DOI: `10.1109/COMST.2018.2814571`. [Online]. Available: `https://ieeexplore.ieee.org/document/8314121/`.

[14] Sandvine, *Internet Phenomena Report*. [Online]. Available: `https://www.sandvine.com/global-internet-phenomena-report-2019`.

[15] J. Summers, T. Brecht, D. Eager, and A. Gutarin, "Characterizing the workload of a netflix streaming video server", *in 2016 IEEE International Symposium on Workload Characterization (IISWC)*, Providence, RI, USA: IEEE, Sep. 2016, pp. 1–12, ISBN: 978-1-5090-3896-1. DOI: `10.1109/IISWC.2016.7581265`. [Online]. Available: `http://ieeexplore.ieee.org/document/7581265/`.

[16] Akamai, *Akamai facts and figures*, 2020. [Online]. Available: `https://www.akamai.com/us/en/about/facts-figures.jsp`.

[17] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: a platform for high-performance internet applications", *ACM SIGOPS Operating Systems Review*, vol. 44, *3*, pp. 2–19, Aug. 2010, ISSN: 0163-5980. DOI: `10.1145/1842733.1842736`. [Online]. Available: `https://dl.acm.org/doi/10.1145/1842733.1842736`.

[18] A. Banerjee, N. Sastry, and C. M. Machuca, "Sharing Content at the Edge of the Network Using Game Theoretic Centrality", *in 2019 21st International Conference on Transparent Optical Networks (ICTON)*, Angers, France: IEEE, Jul. 2019, pp. 1–4, ISBN: 978-1-72812-779-8. DOI: `10.1109/ICTON.2019.8840576`. [Online]. Available: `https://ieeexplore.ieee.org/document/8840576/`.

[19] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN", *ACM SIGCOMM Computer Communication Review*, vol. 48, *1*, pp. 28–34, Apr. 2018, ISSN: 0146-4833. DOI: `10.1145/3211852.3211857`. [Online]. Available: `https://dl.acm.org/doi/10.1145/3211852.3211857`.

[20] F. Bentley, C. Luvogt, M. Silverman, R. Wirasinghe, B. White, and D. Lottridge, "Understanding the Long-Term Use of Smart Speaker Assistants", *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, *3*, pp. 1–24, Sep. 2018, ISSN: 2474-9567, 2474-9567. DOI: `10.1145/3264901`. [Online]. Available: `https://dl.acm.org/doi/10.1145/3264901`.

[21] D. Caputo, L. Verderame, A. Merlo, A. Ranieri, and L. Caviglione, "Are you (Google) Home? Detecting Users' Presence through Traffic Analysis of Smart Speakers", p. 14,

[22] E. E. Dikel, Y. E. Li, M. Vuotari, and S. Mancini, "Evaluating the standby power consumption of smart LED bulbs", *Energy and Buildings*, vol. 186, pp. 71–79, Mar. 2019, ISSN: 03787788. DOI: `10.1016/j.enbuild.2019.01.019`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0378778818333838`.

[23] N. K. Suryadevara and G. R. Biswal, "Smart Plugs: Paradigms and Applications in the Smart City-and-Smart Grid", *Energies*, vol. 12, *10*, p. 1957, May 2019, ISSN: 1996-1073. DOI: `10.3390/en12101957`. [Online]. Available: `https://www.mdpi.com/1996-1073/12/10/1957`.

[24] Google, *Google Nest Thermosta Datasheet*. [Online]. Available: `https://storage.googleapis.com/nest-public-downloads/press/documents/nest-thermostat-fact-sheet_2017.pdf`.

[25] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and classifying IoT traffic in smart cities and campuses", *in 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Atlanta, GA: IEEE, May 2017, pp. 559–564, ISBN: 978-1-5386-2784-6.

DOI: 10.1109/INFCOMW.2017.8116438. [Online]. Available: `http://ieeexplore.ieee.org/document/8116438/`.

[26] L. M. Borges, F. J. Velez, and A. S. Lebres, "Survey on the Characterization and Classification of Wireless Sensor Network Applications", *IEEE Communications Surveys & Tutorials*, vol. 16, *4*, pp. 1860–1890, 2014, ISSN: 1553-877X. DOI: `10.1109/COMST.2014.2320073`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6805127`.

[27] Y. Kabalci, "A survey on smart metering and smart grid communication", *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 302–318, May 2016, ISSN: 13640321. DOI: `10.1016/j.rser.2015.12.114`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1364032115014975`.

[28] A. E. Kouche, "Towards a wireless sensor network platform for the Internet of Things: Sprouts WSN platform", *in 2012 IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 632–636. DOI: `10.1109/ICC.2012.6364196`.

[29] G. B. Fioccola, R. Sommese, I. Tufano, R. Canonico, and G. Ventre, "Polluino: An Efficient Cloud-based Management of IoT Devices for Air Quality Monitoring", p. 6, 2016.

[30] M. Pavani and P. Rao, "Real time pollution monitoring using Wireless Sensor Networks", *in 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada: IEEE, Oct. 2016, pp. 1–6, ISBN: 978-1-5090-0996-1. DOI: `10.1109/IEMCON.2016.7746315`. [Online]. Available: `http://ieeexplore.ieee.org/document/7746315/`.

[31] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of Wireless Sensor Networks: An Up-to-Date Survey", *Applied System Innovation*, vol. 3, *1*, p. 14, Feb. 2020, ISSN: 2571-5577. DOI: `10.3390/asi3010014`. [Online]. Available: `https://www.mdpi.com/2571-5577/3/1/14`.

[32] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, and D. Pfisterer, "SmartSantander: IoT experimentation over a smart city testbed", *Computer Networks*, vol. 61, pp. 217–238, Mar. 2014, ISSN: 13891286. DOI: `10.1016/j.bjp.2013.12.020`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1389128613004337`.

[33] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating Critical Security Issues of the IoT World: Present and Future Challenges", *IEEE Internet of Things Journal*, vol. 5, *4*, pp. 2483–2495, Aug. 2018, ISSN: 2327-4662, 2372-2541. DOI: `10.1109/JIOT.2017.2767291`. [Online]. Available: `https://ieeexplore.ieee.org/document/8086136/`.

[34] H. Aftab, K. Gilani, J. Lee, L. Nkenyereye, S. Jeong, and J. Song, "Analysis of identifiers on IoT platforms", *Digital Communications and Networks*, May 2019, ISSN: 23528648. DOI: `10.1016/j.dcan.2019.05.003`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S2352864818300671`.

[35] C. H. Liu, B. Yang, and T. Liu, "Efficient naming, addressing and profile services in Internet-of-Things sensory environments", *Ad Hoc Networks*, vol. 18, pp. 85–101, Jul. 2014, ISSN: 15708705. DOI: `10.1016/j.adhoc.2013.02.008`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1570870513000280`.

[36] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues", *Computer Networks*, vol. 144, pp. 17–39, Oct. 2018, ISSN: 13891286. DOI: `10.1016/j.comnet.2018.07.017`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1389128618305243`.

[37] A. Gupta, R. Christie, and R. Manjula, "Scalability in Internet of Things: Features, Techniques and Research Challenges", p. 12,

[38] D. Garcia-Carrillo and R. Marin-Lopez, "Lightweight CoAP-Based Bootstrapping Service for the Internet of Things", *Sensors*, vol. 16, *3*, p. 358, Mar. 2016, ISSN: 1424-8220. DOI: `10.3390/s16030358`. [Online]. Available: `http://www.mdpi.com/1424-8220/16/3/358`.

[39] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A Tutorial on IEEE 802.11ax High Efficiency WLANs", *IEEE Communications Surveys & Tutorials*, vol. 21, *1*, pp. 197–216, 2019, ISSN: 1553-877X, 2373-745X. DOI: `10.1109/COMST.2018.2871099`. [Online]. Available: `https://ieeexplore.ieee.org/document/8468986/`.

[40] T. A. M. Phan, J. K. Nurminen, and M. Di Francesco, "Cloud Databases for Internet-of-Things Data", *in 2014 IEEE International Conference on Internet of Things(iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, Taipei, Taiwan: IEEE, Sep. 2014, pp. 117–124, ISBN: 978-1-4799-5967-9. DOI: `10.1109/iThings.2014.26`. [Online]. Available: `http://ieeexplore.ieee.org/document/7059651/`.

[41] L. L. Dhirani, T. Newe, and S. Nizamani, "Can IoT escape Cloud QoS and Cost Pitfalls", *in 2018 12th International Conference on Sensing Technology (ICST)*, Limerick: IEEE, Dec. 2018, pp. 65–70, ISBN: 978-1-5386-5147-6. DOI: `10.1109/ICSensT.2018.8603570`. [Online]. Available: `https://ieeexplore.ieee.org/document/8603570/`.

[42] J. Ni and X. Bai, "A review of air conditioning energy performance in data centers", *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 625–640, Jan. 2017, ISSN: 13640321. DOI: `10.1016/j.rser.2016.09.050`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S136403211630541X`.

[43] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United States data center energy usage report", 2016.

[44] T. G. Grid, "The Green Grid Data Center Power Efficiency Metrics: PUE and DCiE", Tech. Rep., 2007.

[45] J. Shuja, K. Bilal, S. A. Madani, M. Othman, R. Ranjan, P. Balaji, and S. U. Khan, "Survey of Techniques and Architectures for Designing Energy-Efficient Data Centers", *IEEE Systems Journal*, vol. 10, *2*, pp. 507–519, Jun. 2016, ISSN: 1932-8184, 1937-9234, 2373-7816. DOI: `10.1109/JSYST.2014.2315823`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6856153`.

[46] R. V. Aroca and L. M. G. Gonçalves, "Towards green data centers: A comparison of x86 and ARM architectures power efficiency", *Journal of Parallel and Distributed Computing*, vol. 72, *12*, pp. 1770–1780, Dec. 2012, ISSN: 07437315. DOI: `10.1016/j.jpdc.2012.08.005`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0743731512002122`.

[47] G. von Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in DVFS-enabled clusters", *in 2009 IEEE International Conference on Cluster Computing and Workshops*, New Orleans, LA, USA: IEEE, 2009, pp. 1–10, ISBN: 978-1-4244-5011-4. DOI: `10.1109/CLUSTR.2009.5289182`. [Online]. Available: `http://ieeexplore.ieee.org/document/5289182/`.

[48] J. Samual and M. Hussin, "Analysis of DVFS Technique for Efficient - Energy Management in Cloud Data Center", *International Journal of Engineering*, p. 6,

[49] J. Krzywda, A. Ali-Eldin, T. E. Carlson, P.-O. Östberg, and E. Elmroth, "Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling", *Future Generation Computer Systems*, vol. 81, pp. 114–128, Apr. 2018, ISSN: 0167739X. DOI: `10.1016/j.future.2017.10.044`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0167739X17304910`.

[50] F. AL-Hazemi, J. Lorincz, and A. F. Y. Mohammed, "Minimizing Data Center Uninterruptable Power Supply Overload by Server Power Capping", *IEEE Communications Letters*, vol. 23, *8*, pp. 1342–1346, Aug. 2019, ISSN: 1089-7798, 1558-2558, 2373-7891. DOI: `10.1109/LCOMM.2019.2919717`. [Online]. Available: `https://ieeexplore.ieee.org/document/8725570/`.

[51] M. Ranjbari and J. Akbari Torkestani, "A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers", *Journal of Parallel and Distributed Computing*, vol. 113, pp. 55–62, Mar. 2018, ISSN: 07437315. DOI: `10.1016/j.jpdc.2017.10.009`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S074373151730285X`.

[52] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey", *Computer Networks*, vol. 52, *12*, pp. 2292–2330, Aug. 2008, ISSN: 13891286. DOI: `10.1016/j.comnet.2008.04.002`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1389128608001254`.

[53] J. Haimour and O. Abu-Sharkh, "Energy Efficient Sleep/Wake-up Techniques for IOT: A survey", *in 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Amman, Jordan: IEEE, Apr. 2019, pp. 478–484, ISBN: 978-1-5386-7942-5. DOI: `10.1109/JEEIT.2019.8717372`. [Online]. Available: `https://ieeexplore.ieee.org/document/8717372/`.

[54] N. Garg and R. Garg, "Energy harvesting in IoT devices: A survey", *in 2017 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam: IEEE, Dec. 2017, pp. 127–131, ISBN: 978-1-5386-1959-9. DOI: `10.1109/ISS1.2017.8389371`. [Online]. Available: `https://ieeexplore.ieee.org/document/8389371/`.

[55] K. Georgiou, S. Kerrison, Z. Chamski, and K. Eder, "Energy Transparency for Deeply Embedded Programs", *ACM Transactions on Architecture and Code Optimization*, vol. 14, *1*, pp. 1–26, Apr. 2017, arXiv: 1609.02193, ISSN: 1544-3566, 1544-3973. DOI: `10.1145/3046679`. [Online]. Available: `http://arxiv.org/abs/1609.02193`.

[56] D. Hackenberg, R. Schöne, D. Molka, M. S. Müller, and A. Knüpfer, "Quantifying power consumption variations of HPC systems using SPEC MPI benchmarks", *Computer Science - Research and Development*, vol. 25, *3-4*, pp. 155–163, Sep. 2010, ISSN: 1865-2034, 1865-2042. DOI: `10.1007/s00450-010-0118-0`. [Online]. Available: `http://link.springer.com/10.1007/s00450-010-0118-0`.

[57] G. Abbas, Z. Halim, and Z. H. Abbas, "Fairness-Driven Queue Management: A Survey and Taxonomy", *IEEE Communications Surveys & Tutorials*, vol. 18, *1*, pp. 324–367, 2016, ISSN: 1553-877X. DOI: `10.1109/COMST.2015.2463121`. [Online]. Available: `http://ieeexplore.ieee.org/document/7173004/`.

[58] J. Rajendra, D.-M. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", Tech. Rep., 1984.

[59] T. J. Giuli and M. Baker, "Narses: A Scalable Flow-Based Network Simulator", *arXiv:cs/0211024*, Nov. 2002, arXiv: cs/0211024. [Online]. Available: `http://arxiv.org/abs/cs/0211024`.

[60] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms", *Journal of Parallel and Distributed Computing*, vol. 74, *10*, pp. 2899–2917, Oct. 2014, ISSN: 0743-7315. DOI: `10.1016/j.jpdc.2014.06.008`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0743731514001105`.

[61] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing", *Computer*, vol. 40, *12*, pp. 33–37, Dec. 2007, ISSN: 0018-9162. DOI: `10.1109/MC.2007.443`. [Online]. Available: `http://ieeexplore.ieee.org/document/4404806/`.

[62] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell, "Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router", *in Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, IEEE, 2011, pp. 331–336.

[63] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework for Network Devices", *in NETWORKING 2009*, L. Fratta, H. Schulzrinne, Y. Takahashi, and O. Spaniol, Eds., vol. 5550, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 795–808, ISBN: 978-3-642-01398-0 978-3-642-01399-7. DOI: `10.1007/978-3-642-01399-7_62`. [Online]. Available: `http://link.springer.com/10.1007/978-3-642-01399-7_62`.

[64] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for information technology: telecommunications and information exchange between systems : local and metropolitan area networks–specific requirements. Part 11, Part 11,* New York: Institute of Electrical and Electronics Engineers, 2012, OCLC: 796653418, ISBN: 978-0-7381-7245-3. [Online]. Available: `http://ieeexplore.ieee.org/servlet/opac?punumber=6178209`.

[65] E. Budi P., R. A.S., A. Yuwono, and R. Yuwono, *Thermal Noise as Electromagnetic Polutan in Wireless Communication System*, 2015.

[66] P. Fuxjaeger and S. Ruehrup, "Validation of the NS-3 Interference Model for IEEE802.11 Networks", *in 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*, Munich, Germany: IEEE, Oct. 2015, pp. 216–222, ISBN: 978-1-5090-0351-8. DOI: `10.1109/WMNC.2015.40`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7396701`.

[67] S. F. Portegies Zwart, R. G. Belleman, and P. M. Geldof, "High-performance direct gravitational N-body simulations on graphics processing units", *New Astronomy*, vol. 12, *8*, pp. 641–650, Nov. 2007, ISSN: 13841076. DOI: `10.1016/j.newast.2007.05.004`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1384107607000504`.

[68] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous, "Direct code execution: revisiting library OS architecture for reproducible network experiments", *in Proceedings of the ninth ACM conference on Emerging networking experiments and technologies - CoNEXT '13*, Santa Barbara, California, USA: ACM Press, 2013, pp. 217–228, ISBN: 978-1-4503-2101-3. DOI: `10.1145/2535372.2535374`. [Online]. Available: `http://dl.acm.org/citation.cfm?doid=2535372.2535374`.

[69] M. Amoozadeh, B. Ching, C.-N. Chuah, D. Ghosal, and H. M. Zhang, "VENTOS: Vehicular Network Open Simulator with Hardware-in-the-Loop Support", *Procedia Computer Science*, vol. 151, pp. 61–68, 2019, ISSN: 18770509. DOI: `10.1016/j.procs.2019.04.012`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S1877050919304739`.

[70] G. Lauss, M. O. Faruque, K. Schoder, C. Dufour, A. Viehweider, and J. Langston, "Characteristics and Design of Power Hardware-in-the-Loop Simulations for Electrical Power Systems", *IEEE Transactions on Industrial Electronics*, vol. 63, *1*, pp. 406–417,

Jan. 2016, ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2015.2464308. [Online].
Available: http://ieeexplore.ieee.org/document/7177085/.

[71] M. Bacic, "On hardware-in-the-loop simulation", *in Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain: IEEE, 2005, pp. 3194–3198, ISBN: 978-0-7803-9567-1. DOI: 10.1109/CDC.2005.1582653. [Online]. Available: http://ieeexplore.ieee.org/document/1582653/.

[72] Q. Scheitle, M. Wählisch, O. Gasser, T. C. Schmidt, and G. Carle, "Towards an Ecosystem for Reproducible Research in Computer Networking", *in Proceedings of the Reproducibility Workshop on ZZZ - Reproducibility '17*, Los Angeles, CA, USA: ACM Press, 2017, pp. 5–8, ISBN: 978-1-4503-5060-0. DOI: 10.1145/3097766.3097768. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3097766.3097768.

[73] C. Boettiger, "An introduction to Docker for reproducible research", *ACM SIGOPS Operating Systems Review*, vol. 49, *1*, pp. 71–79, Jan. 2015, ISSN: 0163-5980. DOI: 10.1145/2723872.2723882. [Online]. Available: https://dl.acm.org/doi/10.1145/2723872.2723882.

[74] L. Courtès and R. Wurmus, "Reproducible and User-Controlled Software Environments in HPC with Guix", *in Euro-Par 2015: Parallel Processing Workshops*, S. Hunold, A. Costan, D. Giménez, A. Iosup, L. Ricci, M. E. Gómez Requena, V. Scarano, A. L. Varbanescu, S. L. Scott, S. Lankes, J. Weidendorfer, and M. Alexander, Eds., vol. 9523, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 579–591, ISBN: 978-3-319-27307-5 978-3-319-27308-2. DOI: 10.1007/978-3-319-27308-2_47. [Online]. Available: http://link.springer.com/10.1007/978-3-319-27308-2_47.

[75] A. Devresse, F. Delalondre, and F. Schürmann, "Nix based fully automated workflows and ecosystem to guarantee scientific result reproducibility across software environments and systems", *in Proceedings of the 3rd International Workshop on Software Engineering for High Performance Computing in Computational Science and Engineering - SE-HPCCSE '15*, Austin, Texas: ACM Press, 2015, pp. 25–31, ISBN: 978-1-4503-4012-0. DOI: 10.1145/2830168.2830172. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2830168.2830172.

[76] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience*, vol. 41, *1*, pp. 23–

50, Jan. 2011, ISSN: 00380644. DOI: 10.1002/spe.995. [Online]. Available: http://doi.wiley.com/10.1002/spe.995.

[77] P. Velho, L. M. Schnorr, H. Casanova, and A. Legrand, "On the validity of flow-level tcp network models for grid and cloud simulations", *ACM Transactions on Modeling and Computer Simulation*, vol. 23, *4*, pp. 1–26, Oct. 2013, ISSN: 10493301. DOI: 10.1145/2517448. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2556945.2517448.

[78] Xiang Li, Xiaohong Jiang, Kejiang Ye, and Peng Huang, "DartCSim+: Enhanced CloudSim with the Power and Network Models Integrated", *in 2013 IEEE Sixth International Conference on Cloud Computing*, Santa Clara, CA: IEEE, Jun. 2013, pp. 644–651, ISBN: 978-0-7695-5028-2. DOI: 10.1109/CLOUD.2013.53. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6676752.

[79] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments: iFogSim: A toolkit for modeling and simulation of internet of things", *Software: Practice and Experience*, vol. 47, *9*, pp. 1275–1296, Sep. 2017, ISSN: 00380644. DOI: 10.1002/spe.2509. [Online]. Available: http://doi.wiley.com/10.1002/spe.2509.

[80] J. Xie and Y. Deng, "mtCloudSim: A Flow-Level Network Simulator for Multi-Tenant Cloud", *in 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, Wuhan, China: IEEE, Dec. 2016, pp. 332–339, ISBN: 978-1-5090-4457-3. DOI: 10.1109/ICPADS.2016.0052. [Online]. Available: http://ieeexplore.ieee.org/document/7823767/.

[81] G. Karatas, F. Can, G. Dogan, C. Konca, and A. Akbulut, "Multi-tenant architectures in the cloud: A systematic mapping study", *in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya: IEEE, Sep. 2017, pp. 1–4, ISBN: 978-1-5386-1880-6. DOI: 10.1109/IDAP.2017.8090268. [Online]. Available: http://ieeexplore.ieee.org/document/8090268/.

[82] M. Bonaventura and R. Castro, "FLUID-FLOW AND PACKET-LEVEL MODELS OF DATA NETWORKS UNIFIED UNDER A MODULAR/HIERARCHICAL FRAMEWORK: SPEEDUPS AND SIMPLICITY, COMBINED", *in 2018 Winter Simulation Conference (WSC)*, Gothenburg, Sweden: IEEE, Dec. 2018, pp. 3825–3836, ISBN: 978-

1-5386-6572-5. DOI: `10.1109/WSC.2018.8632499`. [Online]. Available: `https://ieeexplore.ieee.org/document/8632499/`.

[83] B. Liu, D. R. Figueiredo, Y. Guo, J. Kurose, and D. Towsley, "A study of networks simulation efficiency: fluid simulation vs. packet-level simulation", *in Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3, 2001, 1244–1253 vol.3. DOI: `10.1109/INFCOM.2001.916619`.

[84] B. Liu, Y. Guo, J. Kurose, D. Towsley, and W. Gong, "Fluid Simulation of Large Scale Networks: Issues and Tradeoffs", p. 7,

[85] S. Barrachina-Munoz, F. Wilhelmi, I. Selinis, and B. Bellalta, "Komondor: a Wireless Network Simulator for Next-Generation High-Density WLANs", *in 2019 Wireless Days (WD)*, Manchester, United Kingdom: IEEE, Apr. 2019, pp. 1–8, ISBN: 978-1-72810-117-0. DOI: `10.1109/WD.2019.8734225`. [Online]. Available: `https://ieeexplore.ieee.org/document/8734225/`.

[86] S. Sharma, A. N. Mahajan, and R. C. Poonia, "An Inclusive survey of Network Simulators", *SSRN Electronic Journal*, 2019, ISSN: 1556-5068. DOI: `10.2139/ssrn.3363025`. [Online]. Available: `https://www.ssrn.com/abstract=3363025`.

[87] R. L. Patel, "Survey on Network Simulators", *International Journal of Computer Applications*, vol. 182, *21*, p. 9,

[88] F. Nizzi and T. Pecorella, "Protocol Prototype Implementation Using ns-3: a Use-Case", *in Proceedings of the 2019 Workshop on Next-Generation Wireless with ns-3*, Florence Italy: ACM, Jun. 2019, pp. 30–33, ISBN: 978-1-4503-7278-7. DOI: `10.1145/3337941.3337950`. [Online]. Available: `https://dl.acm.org/doi/10.1145/3337941.3337950`.

[89] S. Iskounen, T.-M.-T. Nguyen, and S. Monnet, "WiFi-Direct Simulation for INET in OMNeT++", p. 6,

[90] S. Gangadhar, T. A. N. Nguyen, G. Umapathi, and J. P. G. Sterbenz, "TCP Westwood(+) Protocol Implementation in ns-3", p. 9,

[91] K. Mehdi, M. Lounis, A. Bounceur, and T. Kechadi, "CupCarbon: A Multi-agent and Discrete Event Wireless Sensor Network Design and Simulation Tool", *in Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 126–131, ISBN: 978-1-63190-007-5. DOI:

122

10.4108/icst.simutools.2014.254811. [Online]. Available: https://doi.org/10.4108/icst.simutools.2014.254811.

[92] T. J. Schriber and D. T. Brunner, "How Discrete-Event Simulation Software Works", *in Handbook of Simulation*, J. Banks, Ed., Hoboken, NJ, USA: John Wiley & Sons, Inc., Aug. 1998, pp. 765–811, ISBN: 978-0-470-17244-5 978-0-471-13403-9. DOI: 10.1002/9780470172445.ch24. [Online]. Available: http://doi.wiley.com/10.1002/9780470172445.ch24.

[93] C. Lopez-Pavon, S. Sendra, and J. F. Valenzuela-Valdes, "Evaluation of CupCarbon Network Simulator for Wireless Sensor Networks", *Network Protocols and Algorithms*, vol. 10, *2*, p. 1, Jun. 2018, ISSN: 1943-3581. DOI: 10.5296/npa.v10i2.13201. [Online]. Available: http://www.macrothink.org/journal/index.php/npa/article/view/13201.

[94] A. Bounceur, "CupCarbon: A New Platform for Designing and Simulating Smart-City and IoT Wireless Sensor Networks (SCI-WSN)", *in Proceedings of the International Conference on Internet of Things and Cloud Computing*, New York, NY, USA: ACM, 2016, 1:1–1:1, ISBN: 978-1-4503-4063-2. DOI: 10.1145/2896387.2900336. [Online]. Available: http://doi.acm.org/10.1145/2896387.2900336.

[95] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta, "To overlap or not to overlap: Enabling channel bonding in high-density WLANs", *Computer Networks*, vol. 152, pp. 40–53, Apr. 2019, ISSN: 13891286. DOI: 10.1016/j.comnet.2019.01.018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1389128618309745.

[96] A. Varga and R. Hornig, "AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT", *in Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications Networks and Systems*, Marseille, France: ICST, 2008, ISBN: 978-963-9799-23-3. DOI: 10.4108/ICST.SIMUTOOLS2008.3027. [Online]. Available: http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2008.3027.

[97] *The INET Framework Home Page*. [Online]. Available: https://inet.omnetpp.org/.

[98] A. El Ksimi, C. Leghris, and F. Khoukhi, "Minimizing energy consumption in IPv6 for Small Objects Network", *in 2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*, Marrakech, Morocco: IEEE, Sep. 2016, pp. 332–336, ISBN: 978-1-5090-5146-5. DOI: 10.1109/ICMCS.2016.7905534. [Online]. Available: http://ieeexplore.ieee.org/document/7905534/.

[99] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator", *in Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34, ISBN: 978-3-642-12330-6 978-3-642-12331-3. DOI: `10.1007/978-3-642-12331-3_2`. [Online]. Available: `http://link.springer.com/10.1007/978-3-642-12331-3_2`.

[100] H. Wu, S. Nabar, and R. Poovendran, "An Energy Framework for the Network Simulator 3 (ns-3)", *in Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, Barcelona, Spain: ACM, 2011, ISBN: 978-1-936968-00-8. DOI: `10.4108/icst.simutools.2011.245534`. [Online]. Available: `http://eudl.eu/doi/10.4108/icst.simutools.2011.245534`.

[101] B. F. Cornea, A. C. Orgerie, and L. Lefèvre, "Studying the energy consumption of data transfers in Clouds: the Ecofen approach", *in 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, Oct. 2014, pp. 143–148. DOI: `10.1109/CloudNet.2014.6968983`.

[102] A.-C. Orgerie, L. Lefevre, I. Guerin-Lassous, and D. M. Lopez Pacheco, "ECOFEN: An End-to-end energy Cost mOdel and simulator For Evaluating power consumption in large-scale Networks", *in 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Lucca, Italy: IEEE, Jun. 2011, pp. 1–6, ISBN: 978-1-4577-0352-2. DOI: `10.1109/WoWMoM.2011.5986203`. [Online]. Available: `http://ieeexplore.ieee.org/document/5986203/`.

[103] A.-C. Orgerie, B. L. Amersho, T. Haudebourg, M. Quinson, M. Rifai, D. L. Pacheco, and L. Lefèvre, *Simulation Toolbox for Studying Energy Consumption in Wired Networks*, 2017.

[104] P. Velho, L. Schnorr, H. Casanova, and A. Legrand, "Flow-level network models: have we reached the limits?", p. 28,

[105] R. Brown, "Calendar queues: a fast 0(1) priority queue implementation for the simulation event set problem", *Communications of the ACM*, vol. 31, *10*, pp. 1220–1227, Oct. 1988, ISSN: 0001-0782, 1557-7317. DOI: `10.1145/63039.63045`. [Online]. Available: `https://dl.acm.org/doi/10.1145/63039.63045`.

[106] T. Kleiberg, B. Fu, F. A. Kuipers, P. V. Mieghem, S. Avallone, and B. Quoitin, "De-SiNe: a flow-level QoS Simulator of Networks", *in Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications Networks*

and Systems, Marseille, France: ICST, 2008, ISBN: 978-963-9799-23-3. DOI: `10.4108/ICST.SIMUTOOLS2008.3006`. [Online]. Available: `http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2008.3006`.

[107] G. Anggono and T. Moors, "A Flow-Level Extension to OMNeT++ for Long Simulations of Large Networks", *IEEE Communications Letters*, vol. 21, *3*, pp. 496–499, Mar. 2017, ISSN: 1089-7798. DOI: `10.1109/LCOMM.2016.2628356`. [Online]. Available: `http://ieeexplore.ieee.org/document/7742927/`.

[108] H. Casanova, S. Pandey, J. Oeth, R. Tanaka, F. Suter, and R. Ferreira da Silva, "WRENCH: A Framework for Simulating Workflow Management Systems", *in 2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, Dallas, TX, USA: IEEE, Nov. 2018, pp. 74–85, ISBN: 978-1-72810-196-5. DOI: `10.1109/WORKS.2018.00013`. [Online]. Available: `https://ieeexplore.ieee.org/document/8638379/`.

[109] É. Michon, J. Gossa, S. Genaud, L. Unbekandt, and V. Kherbache, "Schlouder: A broker for IaaS clouds", *Future Generation Computer Systems*, vol. 69, pp. 11–23, Apr. 2017, ISSN: 0167739X. DOI: `10.1016/j.future.2016.09.010`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0167739X16303077`.

[110] S. Alonso-Monsalve, F. Garcia-Carballeira, and A. Calderon, "Fog computing through public-resource computing and storage", *in 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, Valencia, Spain: IEEE, May 2017, pp. 81–87, ISBN: 978-1-5386-2859-1. DOI: `10.1109/FMEC.2017.7946412`. [Online]. Available: `http://ieeexplore.ieee.org/document/7946412/`.

[111] F. C. Heinrich, T. Cornebize, A. Degomme, A. Legrand, A. Carpen-Amarie, S. Hunold, A.-C. Orgerie, and M. Quinson, "Predicting the Energy-Consumption of MPI Applications at Scale Using Only a Single Node", IEEE, Sep. 2017, pp. 92–102, ISBN: 978-1-5386-2326-8. DOI: `10.1109/CLUSTER.2017.66`. [Online]. Available: `http://ieeexplore.ieee.org/document/8048921/`.

[112] F. C. Heinrich, T. Cornebize, A. Degomme, A. Legrand, A. Carpen-Amarie, S. Hunold, A.-C. Orgerie, and M. Quinson, "Predicting the Energy-Consumption of MPI Applications at Scale Using Only a Single Node", *in 2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Honolulu, HI, USA: IEEE, Sep. 2017, pp. 92–102, ISBN: 978-1-5386-2326-8. DOI: `10.1109/CLUSTER.2017.66`. [Online]. Available: `http://ieeexplore.ieee.org/document/8048921/`.

[113] T. Hirofuchi, A. Lebre, and L. Pouilloux, "SimGrid VM: Virtual Machine Support for a Simulation Framework of Distributed Systems", *IEEE Transactions on Cloud Computing*, vol. 6, *1*, pp. 221–234, Jan. 2018, ISSN: 2168-7161. DOI: 10.1109/TCC.2015.2481422. [Online]. Available: http://ieeexplore.ieee.org/document/7274701/.

[114] N. McDonald, A. Flores, A. Davis, M. Isaev, J. Kim, and D. Gibson, "SuperSim: Extensible Flit-Level Simulation of Large-Scale Interconnection Networks", *in 2018 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Belfast: IEEE, Apr. 2018, pp. 87–98, ISBN: 978-1-5386-5010-3. DOI: 10.1109/ISPASS.2018.00017. [Online]. Available: https://ieeexplore.ieee.org/document/8366938/.

[115] Sandvine, *The Global Internet Phenomena Report*, https://www.sandvine.com/phenomena, Oct. 2018.

[116] Y. Li, A.-C. Orgerie, I. Rodero, B. L. Amersho, M. Parashar, and J.-M. Menaud, "End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT", *Future Generation Computer Systems*, vol. 87, pp. 667–678, Oct. 2018, ISSN: 0167739X. DOI: 10.1016/j.future.2017.12.048. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X17314309.

[117] Google, *Nest Learning Thermostat – Spec Sheet*, https://nest.com/-downloads/press/documents/nest-thermostat-fact-sheet_2017.pdf, 2017.

[118] Cisco, *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*, White paper, Feb. 2019.

[119] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n Power Consumption", p. 5,

[120] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United States Data Center Energy Usage Report", Tech. Rep. LBNL–1005775, 1372902, Jun. 2016. DOI: 10.2172/1372902. [Online]. Available: http://www.osti.gov/servlets/purl/1372902/.

[121] E. Ahvar, A.-C. Orgerie, and A. Lebre, "Estimating Energy Consumption of Cloud, Fog and Edge Computing Infrastructures", *IEEE Trans. on Sust. Comp.*, 2019.

[122] A. Hassidim, D. Raz, M. Segalov, and A. Shaqed, "Network utilization: The flow view", *in IEEE INFOCOM*, 2013, pp. 1429–1437.

[123] Z. Zhang, Y. Bejerano, and S. Antonakopoulos, "Energy-Efficient IP Core Network Configuration Under General Traffic Demands", *IEEE/ACM Trans. on Networking*, vol. 24, *2*, pp. 745–758, 2016.

[124] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework for Network Devices", *in NETWORKING 2009*, ser. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, May 2009, pp. 795–808, ISBN: 978-3-642-01398-0 978-3-642-01399-7. DOI: `10.1007/978-3-642-01399-7_62`. [Online]. Available: `https://link.springer.com/chapter/10.1007/978-3-642-01399-7_62`.

[125] L. Guegan and A.-C. Orgerie, *Estimating the end-to-end energy consumption of low-bandwidth IoT applications for WiFi devices*, 2019. [Online]. Available: `https://gitlab.inria.fr/lguegan/paper-lowrate-iot`.

[126] C. Fiandrino, D. Kliazovich, P. Bouvry, and A. Y. Zomaya, "Performance Metrics for Data Center Communication Systems", IEEE, Jun. 2015, pp. 98–105, ISBN: 978-1-4673-7287-9. DOI: `10.1109/CLOUD.2015.23`. [Online]. Available: `http://ieeexplore.ieee.org/document/7214033/`.

[127] B. F. Cornea, A. C. Orgerie, and L. Lefèvre, "Studying the energy consumption of data transfers in Clouds: the Ecofen approach", *in 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, Oct. 2014, pp. 143–148. DOI: `10.1109/CloudNet.2014.6968983`.

[128] A.-C. Orgerie, B. L. Amersho, T. Haudebourg, M. Quinson, M. Rifai, D. L. Pacheco, and L. Lefèvre, "Simulation Toolbox for Studying Energy Consumption in Wired Networks", *in CNSM: International Conference on Network and Service Management*, 2017, pp. 1–5.

[129] L. Guegan, B. L. Amersho, A.-C. Orgerie, and M. Quinson, *A Large-Scale Wired Network Energy Model for Flow-Level Simulations*, 2019. [Online]. Available: `https://gitlab.inria.fr/quinson/paper-simgrid-link-energy`.

[130] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers", *The Journal of Supercomputing*, vol. 62, *3*, pp. 1263–1283, Dec. 2012, ISSN: 0920-8542, 1573-0484. DOI: `10.1007/s11227-010-0504-1`. [Online]. Available: `http://link.springer.com/10.1007/s11227-010-0504-1`.

[131] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan, "GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers", *in 2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec. 2010, pp. 1–5. DOI: `10.1109/GLOCOM.2010.5683561`.

[132] M. Y. Robert, M. D. Abramson, M. E. Bampis, and M. M. Snir, "Scheduling for Large Scale Distributed Computing Systems: Approaches and Per- formance Evaluation Issues", p. 179,

[133] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE Journal on selected areas in communications*, vol. 18, *3*, pp. 535–547, 2000.

[134] M. Maity, B. Raman, and M. Vutukuru, "TCP Download Performance in Dense WiFi Scenarios: Analysis and Solution", *IEEE Transactions on Mobile Computing*, vol. 16, *1*, pp. 213–227, Jan. 2017, ISSN: 1536-1233. DOI: `10.1109/TMC.2016.2540632`. [Online]. Available: `http://ieeexplore.ieee.org/document/7430293/`.

[135] H. Arbabi and M. C. Weigle, "Highway mobility and vehicular ad-hoc networks in ns-3", *in Proceedings of the 2010 Winter Simulation Conference*, Baltimore, MD, USA: IEEE, Dec. 2010, pp. 2991–3003, ISBN: 978-1-4244-9866-6. DOI: `10.1109/WSC.2010.5678993`. [Online]. Available: `http://ieeexplore.ieee.org/document/5678993/`.

[136] N. S. Ravindranath, I. Singh, A. Prasad, and V. S. Rao, "Performance Evaluation of IEEE 802.11ac and 802.11n using NS3", *Indian Journal of Science and Technology*, vol. 9, *26*, Jul. 2016, ISSN: 0974-5645, 0974-6846. DOI: `10.17485/ijst/2016/v9i26/93565`. [Online]. Available: `https://indjst.org/articles/performance-evaluation-of-ieee-80211ac-and-80211n-using-ns3`.

[137] M. Abu-Tair and S. N. Bhatti, "Impact of Cell Load on 5GHz IEEE 802.11 WLAN", *in 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Taipei, Taiwan: IEEE, Mar. 2017, pp. 716–721, ISBN: 978-1-5090-6231-7. DOI: `10.1109/WAINA.2017.27`. [Online]. Available: `http://ieeexplore.ieee.org/document/7929770/`.

[138] G. Bhanage, "AMSDU vs AMPDU: A Brief Tutorial on WiFi Aggregation Support", p. 3,

[139] G. Pei and T. R. Henderson, "Validation of OFDM model in ns-3", p. 5,

[140] V. K. Sarker, J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "A Survey on LoRa for IoT: Integrating Edge Computing", *in 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, Rome, Italy: IEEE, Jun. 2019, pp. 295–300, ISBN: 978-1-72811-796-6. DOI: `10.1109/FMEC.2019.8795313`. [Online]. Available: `https://ieeexplore.ieee.org/document/8795313/`.

[141] S. Tahir, S. T. Bakhsh, M. Abulkhair, and M. O. Alassafi, "An energy-efficient fog-to-cloud Internet of Medical Things architecture", *International Journal of Distributed Sensor Networks*, vol. 15, *5*, p. 155 014 771 985 197, May 2019, ISSN: 1550-1477, 1550-1477. DOI: `10.1177/1550147719851977`. [Online]. Available: `http://journals.sagepub.com/doi/10.1177/1550147719851977`.

[142] A. Mebrek, L. Merghem-Boulahia, and M. Esseghir, "Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing", *in 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA: IEEE, Oct. 2017, pp. 1–4, ISBN: 978-1-5386-1465-5. DOI: `10.1109/NCA.2017.8171359`. [Online]. Available: `http://ieeexplore.ieee.org/document/8171359/`.

[143] G. Li, J. Yan, L. Chen, J. Wu, Q. Lin, and Y. Zhang, "Energy Consumption Optimization With a Delay Threshold in Cloud-Fog Cooperation Computing", *IEEE Access*, vol. 7, pp. 159 688–159 697, 2019, ISSN: 2169-3536. DOI: `10.1109/ACCESS.2019.2950443`. [Online]. Available: `https://ieeexplore.ieee.org/document/8887154/`.

[144] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy Efficient Optimization for Computation Offloading in Fog Computing System", *in GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore: IEEE, Dec. 2017, pp. 1–6, ISBN: 978-1-5090-5019-2. DOI: `10.1109/GLOCOM.2017.8254207`. [Online]. Available: `http://ieeexplore.ieee.org/document/8254207/`.

[145] H. O. Hassan, S. Azizi, and M. Shojafar, "Priority, network and energy-aware placement of IoT-based application services in fog-cloud environments", *IET Communications*, vol. 14, *13*, pp. 2117–2129, Aug. 2020, ISSN: 1751-8628, 1751-8636. DOI: `10.1049/iet-com.2020.0007`. [Online]. Available: `https://digital-library.theiet.org/content/journals/10.1049/iet-com.2020.0007`.

[146] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet Of Things requests to minimize latency in hybrid Fog-Cloud computing", *Future Generation Computer Systems*, vol. 111, pp. 539–551, Oct. 2020, ISSN: 0167739X. DOI: `10.1016/`

`j.future.2019.09.039`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0167739X18303327`.

[147] S. Sarkar, R. Wankar, S. N. Srirama, and N. K. Suryadevara, "Serverless Management of Sensing Systems for Fog Computing Framework", *IEEE Sensors Journal*, vol. 20, *3*, pp. 1564–1572, Feb. 2020, ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: `10.1109/JSEN.2019.2939182`. [Online]. Available: `https://ieeexplore.ieee.org/document/8822951/`.

[148] M. M. Mahmoud, J. J. Rodrigues, K. Saleem, J. Al-Muhtadi, N. Kumar, and V. Korotaev, "Towards energy-aware fog-enabled cloud of things for healthcare", *Computers & Electrical Engineering*, vol. 67, pp. 58–69, Apr. 2018, ISSN: 00457906. DOI: `10.1016/j.compeleceng.2018.02.047`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0045790618300399`.

[149] A. Toor, S. u. Islam, N. Sohail, A. Akhunzada, J. Boudjadar, H. A. Khattak, I. U. Din, and J. J. Rodrigues, "Energy and performance aware fog computing: A case of DVFS and green renewable energy", *Future Generation Computer Systems*, vol. 101, pp. 1112–1121, Dec. 2019, ISSN: 0167739X. DOI: `10.1016/j.future.2019.07.010`. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S0167739X19310234`.

Titre : Modèles scalable pour la prédiction temporelle et énergétique des infrastructures Fog

Mot clés : simulation,réseaux,énergie,modèle,simulateur,SimGrid,Wi-Fi,paquet,flux

Résumé : L'informatique géo-distribuée (Fog Computing) désigne la migration des ressources de calcul et de stockage du nuage (Cloud) vers les utilisateurs. Cette migration des ressources permet de réduire la latence des terminaux utilisateurs afin de répondre à l'évolution des usages de l'Internet. En parallèle, le nombre de terminaux ne cesse de croître avec le développement de l'Internet des objets. Cette croissance des infrastructures et du nombre d'objets connectés à Internet entraîne une hausse de la consommation électrique globale liée au numérique. Cependant, cette consommation est très distribuée et fait intervenir de multiples acteurs : objets connectés, réseaux locaux, fournisseurs d'accès à Internet, infrastructures de Fog et de Cloud. Il est ainsi difficile d'étudier l'impact de la croissance du nombre d'objets connectés sur la consommation électrique des infrastructures qui constituent l'Internet des objets. L'objectif de cette thèse est de proposer des modèles afin de permettre l'étude à grande échelle de la consommation énergétique des infrastructures Fog de manière efficace et reproductible. Les modèles proposés ont été intégrés à l'outil de simulation SimGrid afin d'être validés et diffusés.

Title: Scalable end-to-end models for the time and energy performance of Fog infrastructures

Keywords: simulation,network,energy,model,simulator,SimGrid,Wi-Fi,packet,flow

Abstract: Fog Computing designates the migration of the computing and storage resources of the Cloud towards the edge of the network. This resources migration allows to reduce the user's nodes latency to answer to the evolution of the Internet usages. In parallel, the number of terminal is increasing with the development of the Internet Of Things. This infrastructures growth leads to an increase of the global energy consumption related to network infrastructures. However, this energy consumption is distributed and involved many actors such as: connected objects, local network, Internet Service Providers, Fog and Cloud infrastructures. Thus, it is difficult to study the impact of the connected objects growth on the infrastructures that composed the Internet of Things. The goal of this thesis is to propose models to study the energy consumption of large-scale Fog infrastructures in an efficient and reproducible manner. The proposed models have been integrated in the SimGrid simulation framework in order to be validated and spread.