Lappeenranta University of Technology
School of Engineering Science

Erasmus Mundus Master of Science in
Pervasive Computing and Communications for Sustainable Development

Mina Rady

Budget of IoT Low Power Wide Area Network Architectures

Master’s Thesis
2018

Supervised by:
Professor Francis Lepage, University of Lorraine
Associate Professor Jean-Philippe Georges, University of Lorraine

Examiners:
Professor Eric Rondeau, University of Lorraine
Professor Jari Porras, Lappeenranta University of Technology
Associate Professor Karl Andersson, Luleå University of Technology.
This thesis is prepared as part of an European Erasmus Mundus program
Pervasive Computing & Communications for Sustainable Development

This thesis has been accepted by partner institutions of the consortium (cf. UDL-DAJ, n°1524, 2012 PERCCOM agreement).

Successful defense of this thesis is obligatory for graduation with the following national diplomas:

- Master in Master in Complex Systems Engineering (University of Lorraine)
- Master of Science in Software Engineering (Lappeenranta University of Technology)
- Degree of Master of Science (120 credits) –Major: Computer Science and Engineering,
  Specialization: Pervasive Computing and Communications for Sustainable Development
  (Luleå University of Technology)
ABSTRACT

Lappeenranta University of Technology
School of Engineering Sciences
Degree Program in Computer Science

Mina Rady

Title of the work - Budget of IoT Low Power Wide Area Network Architectures

Master’s Thesis

80 pages, 38 figures, 6 tables, 41 formulas.

Examiners: Professor Eric Rondeau
            Professor Jari Porras
            Assoc. Professor Karl Andersson

Keywords: LPWAN, IoT, M2M, Planning, Smart City

In this thesis, we propose a model for the total budget of IoT LPWAN architectures to estimate their real economic and environmental costs. Based on a system engineering view of an IoT sensor node we provide a comprehensive model that estimates the total operational expenditure (OpEx) of generally any network of sensor nodes while taking into account the variation in technological parameters. We also show that non-radio components may interfere with network Quality of Service (QoS) and we provide verified theoretical framework for accurately predicting and controlling Internet of Things (IoT) node behavior. We provide an optimization model that is guaranteed to find least OpEx-expensive link assigned in an LPWAN IoT connected-star topology with heterogeneous End Device (ED) configurations. We also show that significant budget and environmental hazardous waste savings can be achieved through seemingly passive network changes such as introducing few gateways (GWs) or removing an unneeded timestamp from packet payload.
Acknowledgment

The research reported here was supported and funded by the PERCCOM Erasmus Mundus Program of the European Union (PERCCOM- FPA 2013-0231). The authors would like to express their gratitude to all the partner institutions, sponsors, and researchers involved in the PERCCOM program [1].

We wish to acknowledge the valuable feedback from Mr. Régis Lerbour, Director, Planning & Optimization Product Management at InfoVista, Paris, France who contributed greatly to the maturity of this research.

Furthermore, we wish to acknowledge the contribution of Johan Angot, undergraduate intern at CRAN for his effort in the experiments part of this thesis.
## Contents

1 Introduction  
   1.1 Related Work in LPWAN Performance Analysis and Simulation  12  
   1.2 Background on LPWAN Economy  13  
   1.3 Side-Effects of LPWAN Economic Model  14  
      1.3.1 Limitations on Technical Maturity  14  
      1.3.2 End to End Principle Application Challenge  15  
   1.4 Objectives and Contributions  15  

2 Theoretical Framework  17  
   2.1 Sensor Node as an Engineering System  18  
   2.2 Time Cost Model: Separation Principle  19  
      2.2.1 Radio Transmission Time Cost  21  
      2.2.2 Computation Time Cost: Revisiting Complexity Theory  22  
      2.2.3 Formalization of Time Cost Per Input Element Ratio  23  
      2.2.4 Sensing Time Cost  24  
      2.2.5 IO Variant Time Cost  24  
   2.3 Energy Cost Model: Separation Principle  25  
      2.3.1 Sensing Energy Cost  25  
      2.3.2 Radio Transmission Energy Cost  25  
      2.3.3 Computational Energy Cost  26  
      2.3.4 I/O Energy Cost  26  
   2.4 Network Architecture Budget Model  27  
   2.5 Environmental Cost Model  32  
   2.6 Network Budget Optimization Model  33  

3 Time Cost Model Verification  35  
   3.1 Transmission Time Cost  35  
   3.2 Sensor Variant Time Cost  35  
   3.3 SD Memory IO Time Cost  36  
   3.4 Computational Complexity Time Cost  37
List of Figures

1 Hierarchical Theoretical Framework ........................................ 17
2 Simplified Physical System Diagram of an IoT Sensor System .......... 18
3 Simplified Logical System Diagram of a Sensor System .................. 19
4 Simplified Logical System Diagram of a Sensor System .................. 20
5 Model for estimating link OpEx ............................................. 27
6 Model for estimating ED Chemical Waste and Toxicity .................... 34
7 Measured Bit ToA for different LoRa Spread Factors ...................... 35
8 Measured warmup durations for CO2 and THP sensors ..................... 36
9 Change in Duration of an SD IO Operation for Different Stream Sizes .... 36
10 Change in Duration of an SD IO Time Per Byte ............................ 37
11 Averaging Operation Time with Different Input Sizes and Algorithmic Complexities 38
12 Average processing time per byte with complexities: O(N), O(N^2) and O(N^3) 38
13 Theoretical ratio of processor clocks per array element for an averaging operation 39
14 Simple counting program ................................................... 40
15 Simple counting program with modulo compute overhead .................. 40
16 Measured battery voltage discharge curves for different LoRa spread factors running for a fixed duration .......................... 41
17 Co2 vs THP Sensor energy consumption .................................... 42
18 Impact of Co2 activation toggling every one hour on battery voltage ........ 43
19 IO Battery voltage decrease .................................................. 43
20 Battery discharge curve for various processing loads ....................... 44
21 Impact of different physical layer configurations on battery lifetime and bitrate 45
22 Uncalibrated S0 Node QoS (No sensors attached) ............................ 49
23 Uncalibrated S1 Node (CO2 sensor attached) ................................ 49
24 Calibrated and Synchronized S0 and S1 Packet Arrival Time Trace ........ 49
25 Packet Interarrival Time at Receiver ....................................... 49
26 Network simulation experiment setup ...................................... 53
27 Simulation Framework Outline .............................................. 60
28 Simulation Framework Class Diagram ..................................... 61
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>Estimated Minimum Required Back-haul Throughput on Each Gateway</td>
<td>63</td>
</tr>
<tr>
<td>31</td>
<td>CDF of link OpEx in the network for different coverage densities</td>
<td>63</td>
</tr>
<tr>
<td>30</td>
<td>Impact of coverage density on network budget</td>
<td>64</td>
</tr>
<tr>
<td>32</td>
<td>Impact of ED configuration on network budget</td>
<td>65</td>
</tr>
<tr>
<td>33</td>
<td>CDF of link OpEx in the network for different ED configurations</td>
<td>65</td>
</tr>
<tr>
<td>34</td>
<td>Impact of battery quality on network budget</td>
<td>66</td>
</tr>
<tr>
<td>35</td>
<td>CDF of link OpEx in the network for different battery efficiency levels</td>
<td>67</td>
</tr>
<tr>
<td>36</td>
<td>Impact of LoRa vs. NB-IoT</td>
<td>68</td>
</tr>
<tr>
<td>37</td>
<td>Impact of ILP model ED assignment in heterogeneous battery deployment</td>
<td>70</td>
</tr>
<tr>
<td>38</td>
<td>Sustainability Analysis of Integrated Model</td>
<td>72</td>
</tr>
</tbody>
</table>
List of Tables

1. Nodes S0 and S1 statistics .................................................. 48
2. Cost of reference architecture for 365 Days .......................... 62
3. Savings of each experimental architecture compared to basic configuration deployment ........................................... 69
4. Full OpEx Budget Metrics of All Architectures ...................... 78
5. Energy and Radio Metrics of All Architectures ....................... 79
6. Full Environmental Waste Metrics of All Architectures ............ 80
1 Introduction

LPWAN architectures have been subject to extensive studies to evaluate their general performance limits such as scalability, range, penetration in urban environments and use-case-specific performance limits. Dense deployments of such networks incorporating several thousands is beginning to emerge especially that such devices are expected to reach billions with the penetration of IoT. LPWAN deployments follow star topology or connected-stars topology, which is a departure from the mesh multi-hop topology classically assumed in Wireless Sensor Networks (WSNs) or cellular topology adopted by telecom operators, although LPWAN IoT connectivity is being adopted already by telecom operators. LPWAN PHY technologies rely on extremely low bitrate with long range connectivity reaching several kilometers in urban space or tens of kilometers in suburban space depending on line of sight (LoS) conditions. However, they rely on very low power consumption allowing such devices to provide connectivity in hard to reach areas while remaining on batteries for several years. Being such a nascent technology, state of the art research has reported several empirical accounts of its performance in different experimental settings. However, there is a knowledge gap of generic evaluation model of LPWAN systems to allow benchmarking of different architectural configurations regardless of their technical or contextual setting. The proliferation of IoT LPWAN large-scale deployments such as outlined in [2] creates a demand for metrology to estimate operational costs of LPWAN architectures. Furthermore, the dependability on various sorts of batteries as primary or secondary energy sources (in combination with energy harvesting), creates further demand to evaluate the environmental footprints of such deployments. While such networks are usually deployed in connected-stars topology, planning the assignment of user equipment to gateways is a complex problem given the heterogeneity of operational costs of different user equipment nodes. For instance, if we consider an IoT network of several thousands of nodes in connected-star topology, with each node on different battery capacities, battery costs, battery recharge-cycles (durability), and optionally, operator subscription costs, we are interested in the following questions:

- How can the real cost of the network be estimated given that each node is running an application with different configurations and PHY layer configurations with different sensors attached as well as different sensor sampling frequencies?
- Further more, for such a network, what is the expected environmental footprint of the
network given chemical properties of different batteries installed in the end nodes?

- More importantly, it is interesting to quantify how can changing any of the network, sensing, or application configurations impact the budget or the environmental footprint of the network in terms of chemical hazardous waste?

- And finally, how can nodes be assigned to gateways such that total network cost is minimized while respecting QoS minimum constraints in the network? Such question needs to be answered in the light of estimated propagation loss for each device in the network which is specific to the physical settings of the deployments such as geographical terrain elevation profile.

In such context, planning the network such that the overall operational costs are optimized would be a reasonable objective which we address in this thesis. The building block of our network architecture budget estimation framework is an Operational Expenditure (OpEx) model for an IoT sensor node as a sensor system (SS) based on system modeling approach and we demonstrate the necessity of benchmarking different SS components to allow accurate estimation of SS performance and consequently, predictable and optimal SS performance in energy space and in time space.

In time space, our experiments show that configured End Device (ED) execution cycle time duration may vary significantly from measured effective execution cycle due to time consumption of radio, sensing, I/O, and computational components of the sensor node and sleep duration which need separate bench-marking and modeling. In energy space, our experiments show that estimation of effective energy consumption of a sensor node must take into account, separately, the energy consumption of radio, sensing, I/O, and processing time complexity of running program. Our experiments also show that neglecting such discrepancies (especially in the time domain) may lead to significant network QoS deterioration in terms of packet loss as well as complex behavior in battery discharge rate. The thesis report is organized as follows:

- In chapter 1.1 we present related work and outline our contributions.

- In methodology chapter 2 we introduce theoretical foundations organized as A) Sensor System model in general, then B) Time cost model for SS components and C) Energy cost model. Then based on these models, we introduce D) General network OpEx model,
E) Network environmental cost model, and finally F) Network global budget optimization model.

- In chapters 4 and 3 we show the verification results of Time cost model and Energy cost model for SS components respectively. In section 5 we empirically verify the System Model showing complex effects of non-networking components on network QoS.

- In chapter 7 we present the architecture of the simulation framework which we implemented in Matlab to realize our theoretical framework in an accessible object-oriented paradigm.

- In chapter 6 we outline experiment design and setup to verify OpEx, environmental cost and optimization models performance in a large scale realistic simulation for different network configurations.

- In chapter 8 we demonstrate experiment results and we conclude in section 9

1.1 Related Work in LPWAN Performance Analysis and Simulation


Research in [14] is one of the most cited papers in LoRaWAN until the writing of this report. It describes the theoretical limits of LoRaWAN based on Time on Air (ToA) calculations for transmissions with varying, LoRa Spread Factors (SFs), payload sizes, number of EDs, and ED packet generation rate. The paper presents calculated upper limitations of LoRa RF capacity and stresses the impeding role of duty cycle that can contribute to deteriorated QoS in the network because the main LoRaWAN’s reliability is essentially based on Gateway (GW) acknowledgments and GWs must also respect the duty cycle regulations (often 1%) [6]. Additional research in [15] proposes NS3 simulation extension for LoRaWAN which provides QoS estimations based LoRaWAN characteristics.
There is a common knowledge gap in this literature of generic metrics for reporting energy-wise performance of EDs or of whole architectures such that different deployments can be benchmarked irrespective of the details of experimental setup. For instance, it cannot be deduced from most experiments how device battery lifetime can perform with different PHY configurations or computational configurations. This is a significant gap in the broader context of assessment of the budget of LPWAN architectures as power supply (in the form of battery energy) is the most critical element of financial cost and performance constraint in LPWAN architecture.

Another common knowledge gap is in understanding ED energy and delay performance when different sensor node components with different electronic properties are activated. It is unknown as well if or how internal components can interfere with each other’s performance. Furthermore, application of end to end principle is an unexplored gap in LPWAN architectures [16] as the real capacity and cost of ED-based computation is not explored in any of the cited papers which can shed light on possible energy saving by reducing transmissions and shifting computation when possible to the ED.

1.2 Background on LPWAN Economy

Network cost is comprised of OpEx and Capital Expenditure (CapEx). OpEx may vary dynamically based on deployment configuration especially in dense deployments spanning hundreds or thousands of nodes. Therefore it is a complex and changing characteristic during network operation. However, hardware CapEx varies by manufacturer and market dynamics and therefore is out of the scope of this study. We focus on OpEx estimation assuming a fixed CapEx throughout our analysis.

Batteries are essential incurring cost element of LPWAN OpEx and environmental footprint since LPWAN is essentially battery powered. Lithium-Ion (Li-Ion) rechargeable batteries such as [17] are often relied upon as energy source. However they place environmental toxicity burden as research in [18] estimates different categories of ecotoxicity of Li-Ion batteries due to their internal metals such as cobalt and copper.

Operator subscription costs are another incurring cost elements in deployments that rely on operator sinks such as NB-IoT or commercial LoRaWAN operators. Existing solutions are available that may relay packets from GW to any standard APIs through WiFi, Ethernet or GSM back haul. However, it is only usable if the network owners have their own data processing
and visualization interface. Otherwise, the alternative is to utilize third party applications which come also at an additional license cost.

Current IoT LPWAN systems are rolled out to the market, mostly, as end to end solutions where sensor data can be obtained using either subscription to an operator’s cloud services (such as SigFox, Senet in the U.S. for LoRaWAN, and LoRIoT) or through purchasing private aggregation gateway and server such as Meshlium gateway for Libelium. In either cases, the cost of the network would include subscription to the application access that is the only way to decode and analyze the received data. LoRa, however, offers more flexibility because of its open source programmable interface which allows access to low level radio functions to intercept and decode packets at network interface level. But it does not offer any standardized functionalities to relay the packets to any standard computing interface through serial port that may allow integration with third party applications or through APIs.

SigFox, acting as an operator, offers licensing subscriptions that allow up to 140 up-link packets per day along with up to four down-link packets per day (including ACK frames). While LoRaWAN’s Class A reliability is founded on the ACK/Retransmit frames, SigFox’s attempt to achieve reliability by transmitting each packet three times at three different robustness level to increase the probability of packet reception, thus placing confidence in a minimal level of environmental friendless to radio transmission.

1.3 Side-Effects of LPWAN Economic Model

The economic model of LPWAN architectures has resulted into limitations on the technical maturity of the technology as well as constrains on the feasibility of the end to end principle in LPWAN deployments as we outline in this subsection.

1.3.1 Limitations on Technical Maturity

At the mean time, there is lack of technical support and documentation for the decoding functions of Waspmote LoRa libraries in a way that makes integration of the devices very challenging. For example, function to Send a radio signal receives a character array that consists of the ASCII representation of the Hex payload. The Send Radio function enforces a validity check routine which iterates on each character making sure it is within the ASCII representation of the Hex values. What documentation does not reveal (and what we found through low-level investigating
the source code) is that this mechanism spreads each Hex value to a full ASCII byte instead of just four bits. Not only is the Hex payload sent using double radio and energy resources, but also the received bytes format is incompatible with support functions such as Hex decoders or decryptors. While it is possible for an engineer to write their code to fix this design behavior, it demonstrates to what extent how the pressure to enforce data access pricing resulted in significant resource waste in such resource critical system. This can be a particular challenge to organizations that prefer to configure and deploy their own solutions to protect their data or to enable advanced configurations for their particular applications.

1.3.2 End to End Principle Application Challenge

As outlined by the IETF in the LPWAN problem definition memo [16], application of end to end principle is still one of the gaps in the LPWAN space that needs to be addressed. While examining the LPWAN systems within our scopes of research, we explored, from time and energy point of view, the impact of shifting computational jobs to the end devices in coherence with end to end principle. This is particularly interesting since frequent energy expensive transmissions of sensor readings may be reduced by fewer transmissions of results of statistical computations on such readings (such as averages, standard deviations, ranges, and means) which are often the final purpose of readings. This allows significant energy saving and also reduces a node’s air time which is a critical resource given the duty cycle limitations.

It is expected that local I/O or compute tasks consume less energy than radio transmission or reception. However, there is a major gap in the literature on the cost of computational or storage tasks in the node. Therefore, utilizing computational resources on the node to decrease network OpEx has never been considered. This gap is reinforced by the lack of support for node local programming to pressure customers to purchase access cloud services to that perform all necessary data analytics.

1.4 Objectives and Contributions

The aim of this research is to provide a framework that helps estimating the operational costs of an LPWAN architecture as well as its environmental footprints. We aim as well at providing an approach to optimize network operational costs without affecting network QoS. In the light of such objectives, our research presents the following novel contributions.
• We formalize a generic OpEx model and environmental model for LPWAN architectures that estimates total network costs and solid waste footprint considering any possible underlying architecture setup or technology, thus allowing objective evaluation of LPWAN architectures. We demonstrate verified performance of the model by experimenting on large-scale simulation of a realistic setup.

• We propose an Integer Linear Programming model that is proven to find optimal ED to GW link assignment solution with global minimal OpEx in the network, regardless of network size or ED heterogeneity.

• We show that algorithmic complexity’s impact on IoT node processing time can be potentially negligible compared to input size. We show that in a certain general case of program complexity, time and energy cost per input element approaches a constant value as input size increases unless modular programming is heavily used. Therefore, there is strong potential for shifting computations to EDs but with careful use of modular programming paradigm such as object orientation.

• We demonstrate that in how significant QoS improvement as well as budget and environmental savings can be achieved without changing transmission configuration. Also, significant budget and environmental savings can be achieved through minor network configuration such as removing a timestamp, adding few GWs, or cutting down sensor sampling frequency.
2 Theoretical Framework

In this section, we formalize the theoretical framework which we use to construct our estimations and optimizations of the budget of a given architecture. Our framework is constructed as follows (as in figure 1):

1. We establish general system model view to represent the architecture any sensor node.

2. We establish logical view of the sensor node operation.

3. Then we present a model for the time cost of each sensor node component followed by model for the energy consumption of each component.

4. We propose the OpEx cost model and the environmental cost model on the energy model.

5. Finally, we propose budget optimization model that considers the parameters of the OpEx and environmental cost models as well as the particulars of the propagation loss model of the architecture being evaluated.

Figure 1: Hierarchy of Theoretical Framework
2.1 Sensor Node as an Engineering System

We propose a system view for an IoT SS to visualize how performance of different SS components may impact consumption of different resources and system cost on the long run. Architecture diagram in figure 2, demonstrates the architecture of an IoT SS to signify some fundamental properties are distinct from traditional computing architectures such as A) Loose coupling of energy source as it allows energy source replacement which may impact system behavior variably, and B) Loose coupling of SS components: sensor array, I/O storage components, radio interfaces such that they can be activated only when required by the running program on the microcontroller. Each component may require different energy consumption rates which may lead to unpredictable battery discharge rates and in turn unpredictable or interrupted ED performance.

We begin analyzing SS behavior by presenting its logical components which constitute its operation in figure 3. This model signifies the key logical authority in controlling device behavior which is the running micro-controller program which orchestrates SS components as required. However, all four components are fueled by a finite pool of independent resources: 1) Energy resources as the watt-hour capacity of the battery, 2) Time resources as ToA allowance of duty cycle regulations in the ISM band, 3) Environmental resources as the chemical components of the Li-Ion battery are become hazardous waste at battery end of its lifecycle, and 4) Radio resources as the finite radio access allowance offered by an LPWAN operator based on the subscription plan for the ED.

In figure 4 we show a 3-D view of the separate resource spaces on which the separate SS components perform to signify that a given SS configuration can impact each resource space
independently. Therefore, it becomes necessary to quantify the behavior of each SS component in each of the three spaces. We benchmark SS components in A) Energy space and B) Time space. Since environmental resources in our analysis are exclusively tied to battery chemical components, we extrapolate to environmental space benchmarking analytically in our budget model.

2.2 Time Cost Model: Separation Principle

Time cost is an essential constraint in SS performance as it can scale up or down energy consumption of any SS component. However, the only expression referring to time cost in the literature is duty cycle which defines the percentage of time a device is transmitting relative to when it is not transmitting. Since it is a relative scale, it cannot express time cost in independent comparable quantities. Therefore, we propose few concepts that enable time cost quantification and that we will use throughout the paper:

- **Execution Cycle (EC)** is a finite set of instructions that the device is programmed to execute in an infinite loop,

- **EC Duration (ECD)** is the time taken for one EC to be fully executed,

- **EC Frequency (ECF)** is the inverse of ECD,

- **Transmission Cycle (TC)**: is the complete packet *reception* process at the GW that starts when receiving channel begins preamble detection and ends when the receiver processes the
packet and is ready to listen for new transmissions,

- **TC Duration (TCD)** is the time taken for a TC to be fully executed, and

- **TC Frequency (TCD)**: the inverse of TCD.

We quantify ECD and TCD as the sum of time consumed by each EC and TC component respectively. That is, $ECD_i = \| \vec{EC}_i \|$ and $TCD_j = \| \vec{TC}_j \|$. As an example for $\vec{EC}$ and $\vec{TC}$, assume a sensor node $i$ that is setup to data and transmit them between two sleeping cycles, the following are the logical components of each $\vec{EC}_i$ vector which can be expressed in the vector in 1:

1. Turn on sensors, $T_{on}$
2. Sleep until sensors warm up, $T_w$
3. Configure radio module for transmission, $T_{rc}$
4. Sensors Off and pre-process frame, $T_{proc}$
5. Transmit , $T_{tx}$
6. Post processing including any local logging, $T_{post}$
7. Turn off sensors and sleep, $T_{off}$

$$EC_i = \begin{bmatrix}
T_{on} \\
T_w \\
T_{rc} \\
T_{proc} \\
T_{tx} \\
T_{post} \\
T_{off}
\end{bmatrix}$$ \hspace{1cm} (1)

Furthermore, if a GW is configured to execute certain process at the reception of a packet, the logical components of $\overrightarrow{T\bar{C}_j}$ vector for GW $j$ can be expressed in 2.

1. GW reception time is $T_{rx}$

2. GW process time of received frame (including led blinking and SD storage), $T_{proc}$

$$\overrightarrow{T\bar{C}_j} = \begin{bmatrix}
T_{rx} \\
T_{proc}
\end{bmatrix}$$ \hspace{1cm} (2)

We propose total time cost model of ED's EC can be quantified in equation 3, assuming tasks are in sequential execution.

$$T_{EC} = T_{proc} + \sum_{i=1}^{n} T_{IO} + \sum_{j=1}^{m} T_{sj} + T_{Tr}$$ \hspace{1cm} (3)

Where $T_{proc}$: is time consumed in processing, $T_{IO}$: is the IO time cost in stream size and bus throughput of IO stream $i$, $T_{sj}$: is the function of sensor warm up time for sensor $j$, and $T_{Tr}$: is the time consumed in transmission.

2.2.1 Radio Transmission Time Cost

LoRa modulation symbol ToA is expressed as in equation 4. Total number of symbols $P_{sym}$ of the packet is determined by the proprietary modulation scheme as outlined in LoRa Design Guide in [19] and the effective time for packet transmission as in equation 5. However, SF and BW are the essential elements in defining $T_{sym}$, and therefore transmission energy consumption per bit. NBIoT, is fairly more complex to estimate its bit ToA since it follows a complex Physical Resource Block (PRB) coordination scheme. However, it can be theoretically estimated as $T_b = 1/R_b$ from
the nominal maximum bitrate of 200 kbps as reported in [20]. In general, $T_{\text{packet}}$ is computed for the specific radio technology, and considering packet header configuration, time consumed in transmission during a certain duration $D$ can be expressed in (6) considering a fixed packet rate $PR$.

$$T_{\text{sym}} = \frac{2^{SF}}{BW}$$

$$T_{\text{packet}} = T_{\text{preamble}} + T_{\text{sym}} \times PL_{\text{sym}}$$

$$T_{Tr} = T_{\text{packet}} \times \frac{D}{PR}$$

2.2.2 Computation Time Cost: Revisiting Complexity Theory

Program time cost is essentially defined by its Big-O logical complexity. But effectively, it is also defined by number of program code instructions generated by the compiler and the number of clock periods (CPs) consumed by each instruction. For example, in Atmel ATMega 1280/2560 family, register sum (ADD), consumes one CP, but subroutine call or return (CALL, RET) consume five CPs each [21]. Therefore, a sum operation encapsulated in a subroutine consumes eleven CPs instead of one CP. Thus if a program of $O(N^i)$ algorithm makes $M$ subroutine with $C$ microcontroller CPs per subroutine CALL/RET, we approximate total CPs as in equation 7. Where $CP_{OO}$ is modularization/object orientation (OO) overhead that is essentially independent of $N^i$ and expressed in equation 8. Processing time then can be estimated as in equation 9 where $CP_f$ is processor CP frequency and $T_{Sleep}$ is the sleep time configuration for one EC. We show based on formalization in the following subsection 2.2.3, that in a program architecture with $M \gg N$ such as OO-dense architecture, program time cost $CP_{Total}$ changes insignificantly with increase in $N$ and time cost per array element, $CP_n$, would decrease exponentially with increase of $N$ regardless of program’s Big-O complexity. We estimate $CP_n$ ratio in figure 13 in the appendix. Thus, modular programming paradigms such as OOP do not come without cost and should be
used with high caution in resource-limited environments such that of IoT.

\[ CP_{Total} = N^i + CP_{OO} \]  \hspace{1cm} (7)

\[ CP_{OO} = C \times M \]  \hspace{1cm} (8)

\[ T_{proc} = \frac{CP_{Total}}{CP_f} \times \frac{D}{ECD} \]  \hspace{1cm} (9)

### 2.2.3 Formalization of Time Cost Per Input Element Ratio

If a program of \( O(N^i) \) algorithm makes \( M \) subroutine with \( C \) microcontroller CPs per subroutine \texttt{CALL/RET}, we approximate total CPs:

\[ CP_{Total} = C_{clcks} \times M_{Calls} + N^i. \]  \hspace{1cm} (10)

And consequently, CPs per element \( CP_n \) of \( N \)-size array can be approximated as:

\[ CP_n = \frac{CP_{Total}}{N} = \frac{C \times M + N^i}{N}. \]

However, is intensive modularization or OOP are used in program architecture such that \( M \gg N \), then \( CP_n \) can be estimated as \( CP_n = \frac{C \times M}{N} \). Assuming \( M \) is independent of \( N \), then \( CP_n \) can be estimated as 11:

\[ CP_n = \frac{A}{N} \]  \hspace{1cm} (11)
where $A$ is a constant representing non-compute processor CPs such as those related to subroutine calls. Therefore, as $N$ grows, change in $CP_{total}$ is not expected to be significant however $CP_n$ is expected to decrease exponentially.

2.2.4 Sensing Time Cost

Sensor sampling time, $T_{si}$ of sensor $S_i$ per certain duration $D$ can be obtained as in equation 12. Where $T_w$ is the time consumed to turn on the sensor at warm up, which depends on sensor electronics, and $T_{on}(n)$ is the time consumed at sampling iteration $n$ within total $N$ sampling iterations per duration $D$. $T_{si}$ can be also estimated in a different approach assuming fixed sample duration $T_{on}$ along with static sampling frequency $F_{ds}$ as in equation 13.

$$T_{si} = \int_{1}^{N} T_w + T_{on}(n) \, dT$$  \hspace{1cm} (12)

$$T_{si} = (T_w + T_{on}) \times \frac{D}{F_s}$$  \hspace{1cm} (13)

2.2.5 IO Variant Time Cost

Similarly, time consumed in the I/O operations,$T_{IO}$, is a function of processor I/O throughput $T_{RW}$ and stream size $S$. Processor I/O throughput can be obtained empirically with running $N$ IO iterations of increasing stream size $1 < S < N$ as in equation 14, where $T(S_i)$ is the measured IO delay of $S_i$ bytes and $\sigma$ is the total I/O bytes transferred during experiment. Finally, total total IO time in duration $D$ can be estimated as in 15 assuming fixed stream size $S$ and static IO frequency $F_{IO}$.

$$T_{RW} = \frac{1}{\sigma} \int_{1}^{N} T(S_i) \, dT$$  \hspace{1cm} (14)
\[ T_{IO} = \left( \frac{S}{T_{RW}} \right) \times \frac{D}{F_{IO}} \]  

(15)

2.3 Energy Cost Model: Separation Principle

Energy consumption of a node \( E_n \) is counted as a superposition of the consumption of: processor, I/O Operations, \( m \) enabled sensors, and transceiver energy consumption as expressed in equation 16.

\[
E_{total} = E_{proc} + \sum_{i=1}^{n} E_{IOi} + \sum_{j=1}^{m} E_{Sj} + E_{Tr} - E_h
\]

(16)

where: \( E_{proc} \): energy consumed in processing, \( E_{IOi} \): energy consumed in IO stream \( i \), \( E_{Sj} \): energy consumed by sensor \( j \), \( E_{Tr} \): energy consumed by radio transmission, and \( E_h \): energy provided by harvesting module if available.

2.3.1 Sensing Energy Cost

We quantify the energy consumption of each activated sensor as a function of time and sensor energy consumption rate. Let battery power decreases as a function of time \( S(t) \) when sensor is attached and activated and the neutral node energy consumption follows function \( B(t) \), and energy consumption of an attached sensor \( S_i \) over time is \( P_{Si}(t) \). Energy consumption of \( S_i \) would be expressed as in equation 17.

\[
P(S_i) = B(t) - S(t)
\]

(17)

2.3.2 Radio Transmission Energy Cost

Energy consumption of a radio interface is defined by two factors A) configured transmission power and B) ToA. First, according to the data sheet for SX1272 LoRa modem, withdrawn current is 18 mA for +7 dBm Tx power and 28 mA for +13 dBm [22]. Unlike LoRa, NB-IoT offers little room for configuration on the PHY level as it relies on simple Quadrature Phase Shift Keying (QPSK) modulation with fixed transmission power of 23 dBm in up-link (UL) as outlined
in 3GPP Rel-13 [23, 24]. The report estimates NB-IoT to survive on a 5 W.h battery for 10 years on a daily transmission rate consisting of 200 bytes per packet. Therefore, we roughly extrapolate energy per bit (EPB) for NB-IoT as in equation 18.

\[ EPB_{NBIoT} = \frac{BatteryCapacity}{BatteryDailyLife \times PacketsPerDay \times BytesPerPacket \times 8} \]  

Equation 18

Second, assuming fixed transmission power and fixed packet size, total ToA of the transmitter will depend on its time cost per bit, \( T_b \) which will be discussed in section 2.2.1. But assuming known \( T_b \), total link EPB can be obtained analytically as in equation 19. Finally, radio communication energy cost can be approximated as in equation 20.

\[ EPB = \frac{T_{packet} \times I_{Tx} \times V_{Supply}}{P_{bytes} \times 8} \]  

Equation 19

\[ E_T = Bits_{UL} \times EPB_{Tx} + Bits_{DL} \times EPB_{Rx} \]  

Equation 20

2.3.3 Computational Energy Cost

We quantify computation energy cost as the product of time consumed by a microcontroller executing a program \( T_{proc} \) and its power load \( P_{mc} \) as in equation 21. Program time cost is discussed in section 2.2.2. Common low power microcontroller family in Arduino architectures such as Atmel ATmega 1280/2560 consume as low as 500\( \mu \)A under 1.8v for 1 MHz clock cycles and 0.1\( \mu \)A in power down mode [21].

\[ E_{proc} = T_{proc} \times P_{mc} \]  

Equation 21

2.3.4 I/O Energy Cost

Similarly, we quantify energy cost for storage I/O operations as a function of IO stream size \( S_{bits} \), by I/O shield throughput \( Th_{IO} \) and its power withdraw \( P_{IO} \) as expressed in equation 22. Time cost is discussed in section 2.2.5. A common SD memory shield is used in Arduino architectures.
Figure 5: Model for estimating link OpEx consumes as lows as 100mA on a 3.3V power supply [25].

\[ E_{\text{proc}} = \left( \frac{S_{\text{bits}}}{Th_{\text{IO}}} \right) \times P_{\text{IO}} \]  

(22)

2.4 Network Architecture Budget Model

OpEx model is in general a superposition of the OpEx of the four components of the SS: radio, processing, sensing, and IO. We consider two main financial cost items at the foundation of the model: energy cost, and operator subscriber cost. Energy cost is dependent on the cost of the battery, cost of battery recharge operation, recharge cycles of the battery, and the battery watt-hour capacity. Operator subscriber cost consists of link subscription cost per IoT ED and link capacity which vary by the technology and the provider. A general layout of the OpEx model is outlined in figure 5.

We present OpEx for a given network architecture as an \( n \times m \) matrix \( O \) where \( n \) is the number
of OpEx parameters computed and $m$ is the number of links in the network in a star topology or a cellular network topology. OpEx model estimates OpEx vector $\vec{O}_i$ as function of three parameter vectors: empirical vector $\vec{\alpha}$, simulation vector $\vec{\beta}$ and an analytical vector $\vec{\gamma}$:

- **Empirical parameters** $\vec{\alpha}$: describes ED's physical characteristics. This parameter vector allows tuning the model behavior for different LPWAN technologies and configurations. An LPWAN link of a particular technology is established for a certain duration and at the end some metrics are obtained which are used to calibrate the model for that specific link configuration. The calibration metrics vector $\vec{\alpha}$ as expressed in 23 includes generic metrics that can be applied to any network link regardless of underlying technology which is the first step in a technology-independent model.

$$\vec{\alpha} = \begin{bmatrix} E_{bat} \\ B_{pp} \\ b_u \\ b_n \\ P_s \\ P_l \end{bmatrix}$$

(23)

Where:

- $E_{bat}$: Consumed battery power during experiment, W.h
- $B_{pp}$: Average bytes per packet, bytes
- $b_u$: User bits per packet, bits
- $b_n$: Network control header bits per packet, bits
- $P_s$: Total packets sent, packets
- $P_l$: Total packets lost, packets

- **Simulation parameters** $\vec{\beta}$: defines simulation parameters to allow specific estimations for the simulation context. Vector $\vec{\beta}$ expresses those parameters in 24.

$$\vec{\beta} = \begin{bmatrix} Bat_{rc} \\ Bat_{w.h.} \\ Bat_{cycles} \\ O_T \\ P_l \\ B_{pp} \end{bmatrix}$$

(24)
Where:

- $\beta_{Bre}$: Battery replacement cost, $\€$
- $\beta_r$: OpEx estimation range, days
- $\beta_{w.h.}$: Battery capacity, W.h.
- $\beta_{cycles}$: Battery recharge cycles.
- $\beta_t$: Packet inter-arrival time, secs
- $\beta_{bpp}$: Average bytes per packet, bytes

- **Analytical parameters vector** $\vec{\gamma}$: where calculations extrapolate from both vectors $\vec{\alpha}$ and $\vec{\beta}$ to compute general link profile $\vec{\gamma}$ in 25. Generalization metrics in $\vec{\gamma}$ are obtained as function $G(\vec{\alpha}, \vec{\beta})$ is expressed in 25.

$$
\vec{\gamma} = \begin{bmatrix}
Bat_r \\
Bat_{ld} \\
Bat_{Age} \\
E_b \\
E_{bu} \\
B_h \\
R_{bu} \\
R_{bm} \\
R_l
\end{bmatrix}
$$

Where:

- $Bat_r$: Expected battery replacements during OpEx period
- $Bat_{ld}$: Expected battery life, days
- $Bat_{Age}$: Expected battery Age, years
- $E_b$: Energy per bit, J/b (assuming negligible sleep and compute energy)
- $E_{bu}$: Energy per user bit, J/b
- $B_h$: Bits transmission per hour, bits
- $R_{bu}$: User bit ratio
- $R_{bm}$: Management bit ratio
- $R_l$: Bit loss ratio

Finally, OpEx vector, for a given link $i$ is expressed in 27 and is obtained as:
\[ \vec{O}_i = F(\vec{\alpha}_i, G) \]  

(26)

\[
\vec{O}_i = \begin{bmatrix}
O_N \\
O_{us} \\
O_{nm} \\
O_{Op} \\
O_{AS} \\
O_{Com} \\
O_{IO} \\
O_l
\end{bmatrix}
\]  

(27)

Where:

\( O_N \): Net OpEx (Total cost of battery replacements), €

\( O_{us} \): User Service OpEx, €

\( O_{nm} \): Network Management OpEx, €

\( O_{Op} \): OpEx for Operator Costs, €

\( O_{AS} \): OpEx of Active Sensing, €

\( O_{Com} \): OpEx of Computation, €

\( O_{IO} \): OpEx of IO operations, €

\( O_l \): OpEx loss (€)

Essentially, we obtain a cost profile of the ED as in vector \( \vec{C}_i \) which is part of \( \vec{\gamma}_i \) in 28. We can estimate a raw benchmark metric for cost per W.h. from these parameters as in equation 29. We use \( C_{w.h.} \) in combination with energy consumption of each system component to allow estimation of its financial cost along a given OpEx duration. Similarly, we estimate cost per bit transmission that combines operator costs and energy costs to transmit one bit as in equation 30.

\[ \text{Link Cost Profile} \vec{C}_i = \begin{bmatrix}
C_{w.h.} \\
C_{bt}
\end{bmatrix} \]  

(28)
\[ C_{w.h.} = \frac{(B_{\text{cost}} + B_{\text{in}})}{B_{\text{cycles}}}/B_{\text{capacity}} \]  
\[ (29) \]

\[ C_{bt} = E_{pb} \times C_{w.h.} + \frac{\text{LinkSubCost}}{\text{LinkCapacity}} \]  
\[ (30) \]

Finally, an OpEx model can be estimated based on energy and time models as in formula 31.

\[ \text{OpEx}_\text{Net} = (E_{\text{proc}} + \sum_{i=1}^{n} E_{\text{IOi}} + \sum_{i=1}^{m} E_{S_i}) \times C_{w.h.} + \text{Bits}_{\text{total}} \times C_{bt} \]  
\[ (31) \]

Similarly, we can obtain some elementary metrics such as:

- **OpEx per Joule**: which is the OpEx spent on one battery joul, expressed as in 32 as a function of battery price \( B_{\text{Cost}} \), battery recharge cycles (assuming rechargeable battery) \( B_{\text{cycles}} \), battery installation cost \( B_{\text{InstallationCost}} \), and battery wh capacity \( B_{\text{capacity}} \).

\[ \text{OpEx}_{\text{Joul}} = \left( \frac{B_{\text{Cost}}}{B_{\text{cycles}}} + B_{\text{InstallationCost}} \right)/B_{\text{Capacity}} \]  
\[ (32) \]

- **OpEx Waste**: which is OpEx consumed by lost packets in a network link with average bit loss ration \( BLR \), expressed as in 33:

\[ \text{OpEx}_{\text{Waste}} = BLR \times (\text{OpEx}_{\text{Joul}} \times E_{PB} \times \text{Bits}_{\text{day}} \times \text{OpEx}_{\text{Duration}}) \]  
\[ (33) \]

- **OpEx per bit**: which is the OpEx consumed by transmission of one bit which takes into account energy costs and subscriber cost as expressed in 34:

\[ \text{OpEx}_{\text{bit}} = \left( \frac{\text{LinkSubCost}}{\text{LinkSubCapacity}} \right) + \text{OpEx}_{\text{Joul}} \times E_{PB} \]  
\[ (34) \]
2.5 Environmental Cost Model

We propose an environmental cost model that is based on the solid waste hazardous nature of Li-Ion batteries that are the main source of energy for common commercial LPWAN deployments. Existing research provides average estimations of chemical substance waste percentage per Li-Ion battery grams [18]. Then we extend from our OpEx calculations to estimate the amount of batteries and/or battery charge cycles consumed by an ED based on the Energy and Time cost models and battery specifications of capacity, maximum recharge cycles, and weight.

Solid waste estimations can be obtained by the percentages in equation 38. A general outline of how it is computed in terms of ED link profile is seen in figure 6.

Research presents estimations of battery impact on several categories of environmental toxicity: Human Toxicity Potential and Ecotoxicity potential, Terrestrial ecotoxicity, and Freshwater ecotoxicity. However, all toxicity metrics rely on fixed percentages of various metals in Li-Ion batteries. For instance, Copper presents 65.6% of human toxicity potential hazard while Cobalt contributes 79% of freshwater ecotoxicity and 92% in terrestrial ecotoxicity in the ecotoxicity. Therefore, we focus estimating pure grams of chemical waste as a common denominator to all toxicity metrics.

- Battery Expected Lifetime in years: which is the expected age of the battery before disposal, estimated in equation 35

\[
BatteryLife_{\text{time}} = \frac{Bat_{\text{cycles}} \times Bat_{\text{capacity}}}{E_{\text{day}}} \tag{35}
\]

- Waste: is the estimated chemical solid waste in grams resulting from battery consumption as a function of battery weight and the ratio of OpEx estimation period to total battery estimated lifetime, expressed as in equation 36. The estimated waste of each chemical composition i can be obtained by through the average ratio of each substance in Li-Ion batteries as measured in leaching experiments reported in [18] and expressed in battery ingredients vector \( \vec{B} \) in equation 37. The waste vector \( \vec{Waste} \) of each battery chemical ingredient can be expressed as in 38.

\[
Waste_{\text{total}} = Bat_{\text{weight}} \times \frac{OpEx_{\text{duration}}}{BatteryLife_{\text{time}}} \tag{36}
\]
As OpEx matrix $P$ is obtained for all possible links between $n$ EDs and $m$ GWs, there is still room for network tuning in terms of assigning EDs to GWs. Typical link assignment procedure that considers only RSSI at ED may impact network performance regardless of ED configuration efficiency, for example by overloading GWs in good locations. We propose formulation of the link assignment problem as an Integer Linear Programming (ILP) model to be applied on $P$ matrix in equation 40. We minimize the total OpEx of the network and we impose a linear constraint on number of EDs per GW $\leq N$ such that no GW will be assigned more than $N$ EDs. We define

$$f(E, G, P) = \sum_{i=1}^{E} \sum_{j=1}^{G} X_{ij} P_{ij}^{Net}$$  (39)
where $P_{ij}$ is OpEx cost the link between GW$_j$ at the ED$_i$, expressed as $(P_{ij} \in \mathbb{R})$, $E$ is the number of EDs, $G$ is the number of GWs, and $N_j$ is the maximum possible EDs capacity for GW$_j$. With such formalization, we find global optimal solution for a given network architecture as follows.

\[
\min f(E, G, P),
\]

subject to

\[
\sum_{i=1}^{E} x_{ij} \leq N_j \text{ for each GW}_j
\]

\[
\sum_{j=1}^{G} x_{ij} = 1 \text{ for each ED}_i
\]

where $X_{ij} = \begin{cases} 
1 & \text{ED}_i \text{ is assigned to GW}_j \\
0 & \text{otherwise}
\end{cases}$.
3 Time Cost Model Verification

In this section we show the empirical observations to verify the variation in time space performance of various sensor node components: radio transmission, sensing, IO, and compute.

3.1 Transmission Time Cost

In the same experimental setting in subsection 4.1, we measure the time to transmit one bit at each of the LoRa spread factors and we could verify the multiplier effect of SF on ToA for one bit as seen in figure 7. This is coherent with the SF multiplier effect on energy consumption as a consequence of its impact on ToA.

![Figure 7: Measured Bit ToA for different LoRa Spread Factors](image)

3.2 Sensor Variant Time Cost

In this experiment, we run two identical nodes with THP and CO2 respectively and we can observe that they also differ in their time response to turn on commands. Consequently, by applying formula in subsection 2.2.4 on the measurements, we obtain: $T_{CO2} = 1.312 \text{ secs}$ and $T_{THP} = 0.1208 \text{ secs}$. Experiment results plotted in figure 8 in the appendix.
3.3 SD Memory IO Time Cost

In the same experiment in subsection 4.3, we measure the time cost of the SD IO operations. Processing a character array of size up to 4000 bytes may take 60ms. We can measure IO throughput of two sensor nodes (same vendor and model) as 6799.368 bytes/sec and 7138.252 bytes/sec by applying formula in subsection 2.2.5. IO delay results are detailed in figure 9. However, IO duration per byte decreases significantly with the increase of stream size as seen in figure 10.
3.4 Computational Complexity Time Cost

In the same experiment in section 4.4, we measure the time cost of operations that compute the average of a float array at complexities matching the estimated performance model: $\text{Clocks}_{\text{Total}} = CM + RMN$. We verify that measured time cost per array element per call approaches a constant as $N$ increases as seen in figure 12, and it is coherent with our theoretically derived clocks per element per call ratio as in figure 13. More importantly, we can observe that algorithmic complexity does not affect time cost as much as the array size as observed in figure 11. At maximum, averaging a float array with $N = 1400$ and with $O(N^3)$ complexity consumed about 2.87 ms. This is negligible compared to the duration it would have taken to transmit these items to the gateway.
Figure 11: Averaging Operation Time with Different Input Sizes and Algorithmic Complexities

Figure 12: Average processing time per byte with complexities: $O(N)$, $O(N^2)$ and $O(N^3)$
We outline the result of an experiment that proves the significant impact of time cost of non-compute tasks (i.e. additions expressed as a constant which is ignored in algorithmic complexity theory). In figure 14, we outline a simple program to count until 16 million on ATmel Mega processor with 16MHz clock frequency. In figure 15, we show code for the same program but with a minor addition to compute modulo of the index to 1 million so it would print the index value for each millionth iteration. Since this addition is fixed per iteration, regardless of input size, it is considered a constant addition irrelevant of the input growth and therefore theoretically of negligible impact. As expected for the first program, it consumes nearly one second to perform all the iterations. However, for the second program, it consumes nearly nine minutes to finish all execution. This shows that interpreter produced a large set of instructions to compute the modulo calculation that is not taken into account in estimating the actual time cost of the program.
long long int n = 16000000;
for (long long int i = 0; i < n; i++)
{
    c++;
}
printf("%lli", c);

Figure 14: Simple counting program

long long int c = 0;
long long int n = 16000000;
for (long long int i = 0; i < n; i++)
{
    long long int t = i%1000000;
    if (t==0)
    {
        c++;
        printf("%lli", t);
    }
}
printf("%lli", c);

Figure 15: Simple counting program with modulo compute overhead
4 Energy Cost Model Verification

In this section we show the empirical observations to verify the variation in energy space performance of various sensor node components: radio transmission, sensing, IO, and compute.

4.1 Transmission Energy Cost

Empirically, we verify the significant impact of SF on energy rate in the obtained results as link EPB ranges from $6.96 \times 10^{-8} J$ with SF7 to $17.5 \times 10^{-8} J$ with SF12. Those measurements were obtained by running identical devices for exactly the same duration and we can verify the multiplier effect of SF increase on energy consumption and bit ToA. Results are visualized in figure 16. Furthermore, we obtain theoretical EPB for N-BIoT based on equation 18 as $7.17 \times 10^{-8} J$.

Figure 16: Measured battery voltage discharge curves for different LoRa spread factors running for a fixed duration

In a more comprehensive view, we run experiments relying on energy model presented in Semtech LoRa modem design guide [19] to see the impact of main physical layer parameters: Spread Factor, Bandwidth, and Coding Rate on expected battery daily lifetime while taking into account bitrate. Parameters and observations from simulations are aggregated and sorted by Estimated Battery Life in descending order in figure 17. We can conclude that spread factor
has an extreme impact on bitrate and battery life time. Furthermore, at low spread factors, increasing the bandwidth from 250kHz to 500kHz (area highlighted in yellow) causes estimated battery life to increase by almost 4000 days (approximately 10 years). The global highest point of battery life time and bitrate was of the lowest spread factor and the highest bandwidth and the lowest code rate (4/5).

### 4.2 Sensor Variant Energy Cost

We run two identical nodes: one has CO2 gas sensor and the other has Temperature, Humidity, Pressure (THP) sensor. From experiment measurements, we could calculate voltage decrease rate of device with CO2 sensor as $-5.677 \text{ mV/h}$ and with THP sensor as $-0.727 \text{ mV/h}$. In figure 18, we can observe that activating the sensors has significant impact on battery power consumption. We can also see more clearly the impact of CO2 sensor activation toggling every hour on battery discharge curve in figure 19.

![Figure 18: Co2 vs THP Sensor energy consumption](image)

Figure 18: Co2 vs THP Sensor energy consumption
4.3 IO Energy Cost

Similarly, we run an experiment that writes a stream of increasing payload by time. We see that a node running SD IO undergoes total discharge rate $-5.24 \text{ mV}$ which is higher discharge rate compared to neutral discharge curve as observed in figure 20.
4.4 Computation Energy Cost

In this experiment, ED computes the average of an array of changing size and with changing algorithmic complexities: $O(N)$, $O(N^2)$, and $O(N^3)$. We observe that battery discharge rate was not influenced by the size of the input or the complexity of the algorithm as plotted in figure 21. This is because small heap memory allowed a maximum $N = 1400$ which is too small to impact processor run-time with its speed of 16 MHz.

Figure 21: Battery discharge curve for various processing loads
Figure 17: Impact of different physical layer configurations on battery lifetime and bitrate
5 System Model Verification

We demonstrate the interaction between two separate SS components: sensing and radio interface in time space resulting in unpredictable radio access behavior and deteriorated QoS. We show how to control this interaction with system approach leading, instead, to a near deterministic QoS performance.

5.1 Experiment Overview and Results

In a controlled setup, two sensor nodes $S_0$ and $S_1$ were configured with identical sleep cycles to transmit to a SISO GW except $S_1$ had a CO2 sensor that is periodically activated for sampling and sent extra bytes for CO2 readings. This setup resulted in over 15% packet loss of $S_0$ and $S_1$ in a clear systematic pattern. The theory is that accumulation of sensor warm up time cost and addition byte Tx costs resulted in ECF drifting of $S_1$ causing frequent packet collision with $S_0$ at GW. We were able to precisely calibrate $S_1$ with different ECD accounting for its different time resource needs and we achieved synchronization with near zero packet loss. This verifies system model significance in predicting and controlling ED behavior in general especially with deployments with heterogeneous ED configurations.

5.2 Experiment Setup Details

Both transmitters were configured with exactly same configurations. They were manually turned on at separate time difference of approximately 30 secs. They were configured with identical duty cycle vectors. Experiment was run for 250 mins. Both devices sent the following data in a packet:

- Device Serial ID
- Device Node Nr
- Remaining battery voltage,
- Remaining battery percent,
- Local timestamp.

$S_1$ added additional bits for CO2 reading. At packet reception, receiver would store locally the following:
- SNR
- Rx battery level
- Rx timestamp immediately at reception
- Rx timestamp before packet storage to file.
- Complete Hex Frame to robust (upper limit) and the other is the most energy efficient (lower limit).
- Log of restart events or or frame error events (because I doubted that a node can crash and restart)

Over 15% of packets of each transmitter were lost in a clear systematic manner as observed in figures 22 and 23. Introducing the gas sensor increases the ECD of $S_1$ due to sensor warm up time. The accumulation of those shifts create a shift in the CO2 sensor ECD and therefore decreasing its ECF less than ECF of $S_0$. This frequency difference distorts synchronization and leads to packet loss due to conflicting arrival times at the SISO Receiver. To confirm our hypothesis, we synchronize ECFs of both nodes according to the proposed process in subsection 5.3. With fixed sleep times as for seventy seconds each and blink time set as 300 milliseconds (to blink three times, 100 ms each), execution cycle frequencies can be calibrated following this procedure:

- ECD measurement: before running the experiment, each device was set with 2x70 secs sleep cycles. ECD was measured for each device. $S_0$ ECD was 146 secs and $S_1$ ECD was 147 secs (measured on device). Therefore according to EC vectors of $S_0$ and $S_1$ in 41, total ECDs are $\|\Delta t_0\| = 146$ secs and $\|\Delta t_1\| = 147$ secs.

$$\Delta t_0 = \begin{bmatrix} 0 \\ 70 \\ 2 \\ 0 \\ 3 \\ 1 \\ 71 \end{bmatrix}, \Delta t_1 = \begin{bmatrix} 1 \\ 70 \\ 2 \\ 0 \\ 3 \\ 1 \\ 70 \end{bmatrix}$$

(41)
<table>
<thead>
<tr>
<th>Measurement</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total timespan (mins)</td>
<td>921.2</td>
<td>921.2</td>
</tr>
<tr>
<td>Total packets sent</td>
<td>377</td>
<td>377</td>
</tr>
<tr>
<td>Lost packets</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Battery voltage decrease (v)</td>
<td>0.035</td>
<td>(0.462)</td>
</tr>
<tr>
<td>Packet size (bytes)</td>
<td>122</td>
<td>134</td>
</tr>
</tbody>
</table>

Table 1: Nodes S0 and S1 statistics

- ECD calibration: S0 first sleep cycle was changed 71 secs, therefore a total ECD of 147 secs. This should theoretically achieve complete synchronization.

- TCD measurement: both S0 and S1 consumed 3 secs in transmission (known from their EC vectors). Gateway consumed 3 seconds in processing received packet with listening unavailability. Therefore a gap larger than 6 secs is theoretically necessary for ideal synchronization. The receiving node stores complete received frame into local file in SD memory. The components of Rx EC vector $\Delta t_{rEC}$ are: Rx duty cycle (unavailability period) $\overrightarrow{\Delta t_{rx}} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ with $\|\overrightarrow{\Delta t_{rx}}\| = 3$ secs.

- EC Initialization: devices were turned on with a 30 secs gap which is theoretically more than enough for complete packet transfer without loss.

Experiment result is that packet loss is almost non-existent and reception window at Rx is quite stable, except for actual inter-arrival time computation with reference to the internal clock. However, packet reception order was alternated for an unknown reason.

The experiment results in table 1 and visualized in graphs 25 and 24 show that both nodes were perfectly synchronized during the entire experiment. Only one packet was lost due to an unknown reason. Inter-arrival times were measured on both senders and receiver and they were plotted in the diagram below and they exactly the same on both devices. This concludes that calibrating ECF is critical for the network QoS when different nodes have different external sensor configurations.
Figure 22: Uncalibrated S0 Node QoS (No sensors attached)

Figure 23: Uncalibrated S1 Node (CO2 sensor attached)

Figure 24: Calibrated and Synchronized S0 and S1 Packet Arrival Time Trace

Figure 25: Packet Interarrival Time at Receiver
5.3 Sensor System Synchronization General Procedure

Based on the aforementioned experiment, we propose formalize the synchronization procedure that allows calibration of a set of Sensor Systems to successfully transmit in a deterministic manner.

- **ED ECD measurement**: For each device the effective complete ECD is measured. An ECD vector is obtained: $\vec{ECD}$. Then $ECD_{\text{max}}$ is obtained as $\max(\vec{ECD})$.

- **ECF calibration**: To ensure all devices have the same ECD, each device $ED_i$ is extended with auxiliary delay time, $\Delta t_i = (ECD_{\text{max}} - ECD_i)$.

- **GW ECD measurement**: Similarly, maximum possible GW ECD is measured as $TCD_{\text{max}}$.

- **EC Initialization**: To avoid and TCF conflict leading to listener unavailability, an initialization time gap $T_I > TCD_{\text{max}}$ is configured between initialization of devices. This ensures, theoretically, that TCF is completely synchronized, assuming static ECDs and TCDs.
6 Experiments Design and Setup

In this section, we introduce experiment setup to verify the performance of our Budget Model in a realistic complex setting. We use GPS coordinates of approximately 2000 LTE base stations (BSs) of Orange mobile operator in the Paris region through Open Data of Île-de-France [26].

We assume deployment sites contain LoRaWAN EDs with CO2 Gas sensors and electric current sensors. Few BSs are selected as GWs and the remaining will act as ED locations covering Paris region. Our OpEx model is supported with empirically calibrated link profiles for LoRa links at Spread Factors 7-12. Southern half of EDs is assigned low durability Li-Ion batteries with 130 g weight, 27.7 W.h capacity and priced at 6 € with 380 recharge cycles. The northern half has high durability batteries with 50 g weight, 12.24 W.h and priced at 22 € with 500 recharge cycles. Both have same replacement cost.

In the default setup, we assume that all devices transmit electric current sensor readings of some critical equipment every minute and that they sample and store locally CO2 value every hour with 70s sample duration. Packet contains 120 bytes of information including 10 bytes timestamp. Current sensory energy is assumed to be supplied from different source. Then we run three sets of experiments:

- First, we examine the impact of enhancing RF coverage on the budget components of the network. We run three experiments with a basic coverage of 5 GWs, dense coverage with 9 GWs, and extra dense coverage with 13 GWs as visualized in details in figure 26.

- Second, we measure the impact of varying ED behavior through four tests: basic configuration, using half daily packet rate, using half sensing sampling rate, and omitting 10 byte timestamp from the packet payload.

- Third, we examine the impact of using fully low durability or fully high durability batteries for the whole network compared the default 50-50 distribution.

- Finally, in the fourth experiment, we show the theoretical performance with NB-IoT PHY deployment instead of LoRa.

We simulate propagation loss using ITM model [27] and then our model assigns lowest possible SF profile to each link based on ED’s RSSI at GW based on RSSI threshold table in [22] for
energy saving. For each set of experiments, we fix an OpEx duration of 365 days and we plot the normalized estimations for each architecture in seven independent dimensions:

- **Net OpEx (€)**: total OpEx of the network,
- **OpEx Sensing (€)**: total OpEx consumed in sensing activities,
- **OpEx Waste (€)**: Total OpEx of lost packets in the network,
- **Total Energy (W.h.)**: total energy consumed by the network,
- **ToA (years)**: sum of radio time of all network links,
- **Battery Expected Age (years)**: Sum of expected battery life cycle durations of the entire network, and
- **Chemical Waste (gms)**: weight of total network chemical waste at end of OpEx duration.

Moreover, we plot the Cumulative Density Functions (CDF) for OpEx of the network in the following section, and we plot as well the additional savings/expenses incurred by each architecture compared to the reference architecture in table 3.
Figure 26: Network simulation experiment setup
7 Simulation Framework

In order to simulate the different scenarios in the experiment design, we build an experimental simulation framework for this purpose. We present in this chapter an outline of the simulation framework we created in Matlab. The purpose of the framework is to use the basic network topology descriptors: RSSI matrix (between $M$ EDs and $N$ GWs), and different GWs ED capacities. Therefore, it becomes possible for the simulation framework to be integrated in the back-end of an existing simulation tool or be utilized independently. We include this overview of the framework architecture to demonstrate the extent to which our theoretical framework can be extended to be applied to heterogeneous scenarios.

7.1 Framework Outline

The simulation process, outlined in figure 27, is initialized by the following phases:

- General simulation setting: in terms of OpEx duration.

- Wireless network topology setting: in terms of RSSI matrix and GWs ED capacity constraints.

Afterwards, the simulator receives configuration for each link either individually or through configuration profiles for different link components, relying on different helpers classes:

- Application Configuration: in terms using ApplicationHelper.

- Sensor Profile Configuration: in this step, one or more sensors can be added to the ED using SensorProfileHelper.

- Channel Configuration: configures radio channel parameters and control bits length using LinkProfileHelper for LoRa or NbIoT.

- Battery Configuration: configures battery profile parameters for the device using BatteryProfileHelper.

After the network setting is completely populated, the computation phases are initialized:

- For each link between $M$ EDs and $N$ GWs, OpEx and Environmental costs are computed.
- Optimal Network Link Assignment is obtained through Integer Linear Programming model with GW capacity constraints and with ED assignment constraints (i.e. each ED is assigned to exactly $\alpha$ GWs with $\alpha \in \mathbb{N} | 1 \leq \alpha \leq N$).

- Computing final vectors for different network OpEx and environmental cost parameters for the selected network links.

### 7.2 Helper Classes

The simulation framework was implemented in Matlab in an Object Oriented paradigm. We relied on different helper classes to facilitate a meaningful framework (similar to NS3 environment). The class diagram is illustrated in figure 28.

#### 7.2.1 Class SimulationInstance

This class instantiates the three models of the simulation: OpEx Model, Environmental Cost Model and then ILPModel. It accepts as input parameters defined in an object of type SimulationParameters.

#### 7.2.2 Class SimulationParameters

This class instantiates essential parameters required to run the simulation:

- **TimeInterval**: for estimation duration in days (e.g. 365 days).

- **LinkRSSI**: which is a matrix containing the RSSI estimations from $M$ EDs to $N$ GWs. Normally, this matrix would be an exported record from wireless propagation simulation tool.

- **LinksBitLossRatio**: estimated bit loss ratio for each link.

- **EDProfiles**: configuration profile for each ED which is represented by an object of class type EDHelper.

- **GWCapacities**: the upper limit of EDs that each GW can support.
7.2.3 Class EDHelper

This class instantiates configuration settings for an ED in the network. It consists of objects representing different ED components (each with its class helper):

- **ApplicationProfile**: an instance of ApplicationHelper.
- **LinkProfile**: an instance of a technology-specific implementation of virtual class LinkProfileHelper.
- **SensorProfile**: an instance of object of type SensorClassHelper representing sensor behavior.
- **BatteryProfile**: an instance of object of type BatteryProfileHelper which represents battery characteristics.

7.2.4 Class ApplicationHelper

Contains Application settings: **PacketRate**, defined as configured packet inter-arrival time in seconds, and **PayloadSize** in bytes.

7.2.5 Virtual Class LinkProfileHelper

This class contains a generic tool for Link configuration management. It contains the following:

- **LinkProfiles**: a vector of objects of type LinkProfile representing the different possible configurations for a particular physical layer technology (such as different LoRa Spread Factors).

- **ComputeProfile (RSSI)**: a generic method to choose the best link configuration profile for a specific link RSSI. This could mean choosing the least energy consuming configuration for a given minimal QoS requirement based on link RSSI.

- **AddLinkProfile (LinkProfile)**: to dynamically increment the LinkProfile vector if necessary.

7.2.6 Class LinkProfile

This class offers a generic container for link performance parameters:
- **EnergyPerBit** and **BitTimeOnAir**: obtained from empirical measurements
- **UserDataRatio** and **ManagementDataRatio**: the ratio of the user to header bits and vice versa.
- **SubscriberCost**: the monthly subscription cost per ED link
- **SubscriberCapacity**: the monthly megabit link allowance.

### 7.2.7 Class LoRaLinkHelper:

This class is an implementation of **LinkProfileHelper** which contains parameters for six different LoRa link profiles obtained from empirical measurements of Spread Factors 7 to 12, outlined in sections 4.1 and 2.2.1. We assume fixed bandwidth of 125khz and transmission power of 15dB, however, a more sophisticated simulation can populate these profile with more alternatives (e.g. different transmission power levels or bandwidths). Remaining link profile parameters are to

This class implements the method **ComputeLinkProfile(RSSI)** by following the Spread Factor to RSSI interval table outlined in [22] and referenced in [7, 15]. This simulates the adaptive data rate (ADR) mechanism of LoRaWAN which adapts link spread factor depending on signal RSSI (but generally stable RSSI). In an advanced evolution, this feature could include shadowing effect such as the one deployed in the NS3 LoRaWAN simulator in [15].

### 7.2.8 Class NbIoTLinkHelper:

This class contains theoretical EPB parameter computed from NB-IoT performance reports as deduced in section 4.1 and **BitTimeOnAir** estimated based on 200kbps theoretical throughput. This class implements a static **ComputeLinkProfile(RSSI)** method as current NB-IoT deployment uses only static power transmission of $+23dB$. We use two theoretical configurations where IP headers are compressed or not compressed as specified in [24, 28].

### 7.2.9 Class SensorProfileHelper

This class helper contains a vector of objects of class type **SensorProfile** which simulate the sensor configuration and behavior parameters obtained empirically.
7.2.10 Class SensorProfile

This class is a generic container for sensor behavior parameters:

- **Type**: sensor name to be used as reference in the simulation environment (e.g. \( \text{CO}_2 \) or THP).

- **EnergyConsumption**: watt-hour consumption of the sensor such as those obtained in section 4.2.

- **SamplingRate**: the daily sampling frequency as configured in the simulation.

- **SampleDuration**: the configured duration in seconds during which a sensor is on for single sample.

7.3 Class BatteryProfileHelper

This class helper contains a vector of objects of class type `BatteryProfile` which contains performance parameters of different types of batteries.

7.3.1 Class BatteryProfile

This class offers a generic container for battery parameters in terms of performance and chemical compositions:

- **RechargeCyles**: the number of maximum recharge cycles of the battery which is \( > 1 \) for Li-Ion batteries.

- **Weight**: the weight in grams of the battery which is used as a reference to estimate the average amounts of different chemical substances in the battery.

- **Cost**: the price of the battery, which usually varies in relation to durability (i.e. recharge cycles), or energy efficiency (i.e. amperes per mg).

- **InstallationCost**: the cost of installing/re-charging the battery initially and between recharge cycles.

- **EnergyCapacity**: the watt-hour capacity of one recharge cycle of the battery.

- **Cost_per_wh**: is a computed parameter as a function of the previous parameters.
• `<Chemicals%>`: is a vector containing the percentages of different chemicals in the batteries: [Lead, Cobalt, Lithium, Nickel, Thallium, and Copper]. Percentages are used for rechargeable Li-Ion batteries as obtained the research in [18].

This class implements the following methods:

• `Compute_wh_capacity`: which computes Cost_per_wh depending on battery parameters.

• `Compute_human_toxicity`, `Compute_terrestrial_toxicity`, and `Compute_water_toxicity`: which computes human toxicity, terrestrial toxicity and water toxicity factor as described in [18]. An advance development of this section is if the waste recycling region is known (in terms of area and location), then it can allow estimation of the expected toxicity density in the waste destination area.
Figure 27: Simulation Framework Outline
Figure 28: Simulation Framework Class Diagram
8 Results

Cost of reference architecture is outlined in Table 2. In Figure 29 we can see the estimated minimum required back haul throughput on each GW for the backbone network. Varying GW density proved to contribute to network OpEx, as expected, by allowing use of less expensive link configurations to account for radio path loss. Deployment of nine GWs instead of five introduced comprehensive enhancement of link OpEx as illustrated in CDF plot in Figure 31. It also improved network general performance on all financial, environmental, and technical metrics as illustrated in Figure 30 of normalized metrics. Precisely, this coverage enhancement created saving near \( \text{€}12K \) of OpEx, 27 kWh of energy, 46 years of ToA, 1300 years of battery life cycle duration, and 300 grams of chemical waste. However, network became more saturated at nine GWs density as performance is not increased significantly with deployment of thirteen GWs, except of increased ToA and decreased OpEx Waste.

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpEx Net</td>
<td>388,301.20 €</td>
</tr>
<tr>
<td>OpEx Sensing</td>
<td>30,186.38 €</td>
</tr>
<tr>
<td>OpEx Waste</td>
<td>-48,675.7531 €</td>
</tr>
<tr>
<td>User Traffic OpEx</td>
<td>328,271.9109 €</td>
</tr>
<tr>
<td>Management Traffic OpEx</td>
<td>60,029.29 €</td>
</tr>
<tr>
<td>Energy</td>
<td>167.902 kWh</td>
</tr>
<tr>
<td>ToA</td>
<td>80.5 years</td>
</tr>
<tr>
<td>Battery Age</td>
<td>9737.5 years</td>
</tr>
<tr>
<td>Chemical Waste</td>
<td>1707.3 gms</td>
</tr>
</tbody>
</table>

Table 2: Cost of reference architecture for 365 Days
Figure 29: Estimated Minimum Required Back-haul Throughput on Each Gateway

Figure 31: CDF of link OpEx in the network for different coverage densities
In the second set of experiments, ED configuration showed to contribute significantly to network OpEx as illustrated in link OpEx CDF in figure 33 and to general network performance as shown in figure 32. We can observe that increasing packet inter-arrival time to 120 seconds instead of 60 seconds introduced the highest impact of improvement to the network on all metrics such as savings of €179 K in OpEx, 27 kWh of energy, more than 40 years of ToA, and nearly 600 gms of chemical waste. Similarly, cutting sensing sampling rate to be every two hours instead of one introduced significant energy saving of €15K in OpEx. Reducing packet size by 10 bytes by omitting the timestamp, for example, created OpEx saving of almost €30 K.
Figure 32: Impact of ED configuration on network budget

Figure 33: CDF of link OpEx in the network for different ED configurations
In the third set of experiments, we observe significant patterns in the impact of battery characteristics on network metrics. Lowest durability batteries showed to reduce network OpEx as in figure 35 which saved OpEx of nearly € 34 K. However, it leaves higher chemical waste and much shorter battery life-cycle age of the network as in figure 34. On the other hand, high durability battery shows better performance, as expected, in battery age in years and in reduced chemical waste, which comes at the expense of increased OpEx for the entire network, OpEx of sensing activities, and OpEx waste.
In the last experiment, we show network metrics compared to NB-IoT deployment based on estimated theoretical EPB and bit ToA for NB-IoT [24, 28]. The impact of throughput on total network ToA is clearly observed compared to LoRa in figure 36. It is also clear that low robustness of NB-IoT modulation compared to LoRa results in much higher OpEx waste due to packet loss (without considering re-transmits).

Employment of ILP optimization for minimizing network OpEx proved to play a significant role in minimizing network OpEx while inducing sometimes unpredictable side-effects. For instance, using homogeneous battery deployments (whether high or low durability) resulted in additional ToA cost of 2.22 years in the network, and using homogeneous NB-IoT deployment resulted in lower energy consumption yet total lower battery life time duration in the network. This is because using heterogeneous battery deployments or PHY configurations creates a gap between OpEx of the nodes in the network which causes network links of low durable batteries
to be much cheaper than that of high durable batteries even at higher spreading factor. In this case, ILP assignment model gives priority of optimal link assignment to EDs with more expensive OpEx to ensure they are connected to closest GWs while other EDs can still have cheaper links even if they are assigned to further GWs, as illustrated in figure 37, resulting in different link ToAs or battery ages from homogeneous deployments where all nodes have the same link OpEx per Joul. In this particular application, the savings of ToA were more than the additional ToAs in the basic configuration, resulting in lower network ToA in total.

All experiments showed interesting savings patterns from the default basic architecture as listed in table 3. Detailed estimations are available at Appendix 9. Improved significant energy consumption savings were achieved at all experiments, reaching up to 56 kWh by just cutting packet rate to be every two minutes instead of one. Similarly, 31 kWh of savings were achieved by enhancing coverage without changing anything of the network or the battery configurations.
<table>
<thead>
<tr>
<th>Deployment</th>
<th>Savings</th>
<th>OpEx (€)</th>
<th>ToA (years)</th>
<th>Battery Age (years)</th>
<th>Chemical Waste (grams)</th>
<th>Energy (W.h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Durability Battery</td>
<td>33,771.34</td>
<td>2.22</td>
<td>3725.18</td>
<td>-315.98</td>
<td>1239.73</td>
<td></td>
</tr>
<tr>
<td>High Durability Battery</td>
<td>(35,167.70)</td>
<td>2.22</td>
<td>-3878.70</td>
<td>367.93</td>
<td>1239.73</td>
<td></td>
</tr>
<tr>
<td>Half Sensing Frequency</td>
<td>15,093.19</td>
<td>0.00</td>
<td>1670.44</td>
<td>284.82</td>
<td>28278.45</td>
<td></td>
</tr>
<tr>
<td>Half Packet Rate</td>
<td>179,057.41</td>
<td>40.25</td>
<td>3198.33</td>
<td>568.86</td>
<td>55672.80</td>
<td></td>
</tr>
<tr>
<td>No Timestamp</td>
<td>29,842.90</td>
<td>6.71</td>
<td>533.06</td>
<td>94.81</td>
<td>9278.80</td>
<td></td>
</tr>
<tr>
<td>Dense Coverage</td>
<td>11,835.62</td>
<td>46.77</td>
<td>1311.03</td>
<td>308.92</td>
<td>27743.51</td>
<td></td>
</tr>
<tr>
<td>Extra Dense Coverage</td>
<td>14,184.34</td>
<td>53.57</td>
<td>1570.89</td>
<td>347.08</td>
<td>31743.48</td>
<td></td>
</tr>
<tr>
<td>NB-IoT (theoretical)</td>
<td>19,267.13</td>
<td>67.20</td>
<td>2,133.12</td>
<td>419.57</td>
<td>39,741.84</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Savings of each experimental architecture compared to basic configuration deployment.
Figure 37: Impact of ILP model ED assignment in heterogeneous battery deployment
9 Discussion and Conclusion

From a more comprehensive sustainability point of view, we describe how our proposed framework contributes to organizational sustainability in a more holistic perspective. Based on the framework proposed in [29], we show proposed sustainability analysis of our integrated budget and environmental models in figure 38. From sustainability point of view, we can highlight interesting insights from based on results in table 3. From economic sustainability perspective, a simple change such as removing timestamp saves nearly €30K which is equivalent of providing two minimum wage jobs in France at €1498.47/month [30]. Similarly, cutting packet rate into half (by transmitting every two minutes instead of one minute) saves nearly €180K which is equivalent of providing ten jobs. Also, cutting sensing frequency to half, by sampling every two hours instead of every one hour saves nearly €15K which is equivalent of providing one job. From environmental sustainability perspective, we can see significant energy savings reaching up to 56 kwhs by cutting packet rate and nearly 30 kwh by adapting some minor network features such as adding additional four GWs or cutting sensing sampling frequency to half. This is a significant observation since it shows that network radio activity is not the only significant area of optimization for LPWAN development.

Therefore, we achieved the following aforementioned contributions:

- We formalized an OpEx model and environmental model for LPWAN architectures that estimates total network costs and solid waste footprint considering any possible underlying architecture setup or technology, thus allowing objective evaluation of LPWAN architectures. We demonstrate verified performance of the model by experimenting on large-scale simulation of a realistic setup.

- We propose an Integer Linear Programming model that is proven to find optimal ED to GW link assignment solution with global minimal OpEx in the network, regardless of network size or ED heterogeneity.

- We show that algorithmic complexity’s impact on IoT node processing time can be potentially negligible compared to input size. We show that in a certain general case of program complexity, time and energy cost per input element approaches a constant value as input size increases unless modular programming is heavily used. Therefore, there is strong
potential for shifting computations to EDs but with careful use of modular programming paradigm such as object orientation.

- We demonstrate that in how significant QoS improvement as well as budget and environmental savings can be achieved without changing transmission configuration. Also, significant budget and environmental savings can be achieved through minor network configuration such as removing a timestamp, adding few GWs, or cutting down sensor sampling frequency.

Interesting future work based on this thesis can be driven from the need to estimate the budget from operators perspective which would include the estimation of the OpEx allocated for backhaul infrastructure. Such OpEx might include parameters such as the energy cost, number of subscribers, or service capacity used for IoT services in base stations. Furthermore, such
framework can be used to estimate the economic efficiency of network MAC negotiation schemes or development of economy-aware network topology design processes.

In conclusion, the research presented in this paper verifies and validate a budget model for LPWAN architectures which allows to quantify the real financial, environmental, energy, and time costs of a dense LPWAN deployment. According to the principle of separation we are able to benchmark, experimentally, different components of a sensor system to understand its behavior in energy space and time space. After translating our benchmarks into an OpEx model, we are able to observe that significant OpEx savings reaching thousands of euros and energy savings reaching tens of kWhs can be achieved through simple network updates such removing a timestamp from the packet payload, slightly reducing sensing sampling or packet rate, or just by introducing few GWs for better coverage.
References


[29] C. Becker, “Requirements : The Key to Sustainability.”

Appendix I: Detailed Architecture Budget Estimations
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Configuration Deployment</td>
<td>388301.1971</td>
<td>30186.38521</td>
<td>-48.67567531</td>
<td>328271.9109</td>
</tr>
<tr>
<td>Low Durability Battery Deployment</td>
<td>354529.8542</td>
<td>18394.84371</td>
<td>-29.95569634</td>
<td>308123.7597</td>
</tr>
<tr>
<td>High Durability Battery Deployment</td>
<td>423468.8996</td>
<td>41789.26204</td>
<td>-68.02344084</td>
<td>349873.0011</td>
</tr>
<tr>
<td>Half Sensing Frequency Deployment</td>
<td>373208.0045</td>
<td>15093.1926</td>
<td>-48.67567531</td>
<td>328271.9109</td>
</tr>
<tr>
<td>Half Packet Rate Deployment</td>
<td>209243.7912</td>
<td>30186.38521</td>
<td>-24.33783766</td>
<td>164135.9555</td>
</tr>
<tr>
<td>No Timestamp Deployment</td>
<td>358458.2961</td>
<td>30186.38521</td>
<td>-44.61408976</td>
<td>298429.0099</td>
</tr>
<tr>
<td>Dense Coverage (9GWs)</td>
<td>376465.577</td>
<td>30186.38521</td>
<td>-27.50924626</td>
<td>317422.5925</td>
</tr>
<tr>
<td>Extra Dense Coverage (13 GWs)</td>
<td>374116.8563</td>
<td>30186.38521</td>
<td>-22.85487844</td>
<td>315269.5985</td>
</tr>
<tr>
<td>Basic Configuration NBioT</td>
<td>368821.1748</td>
<td>30186.38521</td>
<td>-36.23682165</td>
<td>310415.2238</td>
</tr>
<tr>
<td>ILP Min Chemical Waste</td>
<td>390778.0745</td>
<td>30186.38521</td>
<td>-51.74400589</td>
<td>330542.3819</td>
</tr>
<tr>
<td>ILP Max QoS</td>
<td>391389.8907</td>
<td>30186.38521</td>
<td>-44.93609871</td>
<td>331103.2134</td>
</tr>
<tr>
<td>NBioT</td>
<td>368821.1748</td>
<td>30186.38521</td>
<td>-10.54598468</td>
<td>310415.2238</td>
</tr>
</tbody>
</table>

Table 4: Full OpEx Budget Metrics of All Architectures
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Energy During Opex Period</th>
<th>Time On Air</th>
<th>Battery Expected Age Years</th>
<th>Batteries During Opex Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Configuration Deployment</td>
<td>167902.4954</td>
<td>2538889729</td>
<td>9737.543053</td>
<td>207461.2175</td>
</tr>
<tr>
<td>Low Durability Battery Deployment</td>
<td>166662.7638</td>
<td>2468838228</td>
<td>6012.365218</td>
<td>267961.7545</td>
</tr>
<tr>
<td>High Durability Battery Deployment</td>
<td>166662.7638</td>
<td>2468838228</td>
<td>13616.23888</td>
<td>155685.2299</td>
</tr>
<tr>
<td>Half Sensing Frequency Deployment</td>
<td>139624.0474</td>
<td>2538889729</td>
<td>8067.10232</td>
<td>257195.6732</td>
</tr>
<tr>
<td>Half Packet Rate Deployment</td>
<td>112229.6957</td>
<td>1269444865</td>
<td>6539.212259</td>
<td>301343.602</td>
</tr>
<tr>
<td>No Timestamp Deployment</td>
<td>158623.6954</td>
<td>2327315585</td>
<td>9204.487921</td>
<td>218729.2906</td>
</tr>
<tr>
<td>Dense Coverage (9GWs)</td>
<td>140158.9834</td>
<td>1064081514</td>
<td>8426.515222</td>
<td>244131.3108</td>
</tr>
<tr>
<td>Extra Dense Coverage (13 GWs)</td>
<td>136159.0152</td>
<td>849436078.4</td>
<td>8166.650824</td>
<td>248642.4342</td>
</tr>
<tr>
<td>Basic Configuration NB IoT</td>
<td>128336.2747</td>
<td>5005393.92</td>
<td>7580.972645</td>
<td>254848.9419</td>
</tr>
<tr>
<td>ILP Min Chemical Waste</td>
<td>166709.2678</td>
<td>247244474</td>
<td>10010.49908</td>
<td>213453.5714</td>
</tr>
<tr>
<td>ILP Max QoS</td>
<td>170092.8336</td>
<td>2641601882</td>
<td>10078.66161</td>
<td>208543.0419</td>
</tr>
<tr>
<td>NB IoT</td>
<td>128336.2747</td>
<td>5005393.92</td>
<td>7580.972645</td>
<td>254848.9419</td>
</tr>
</tbody>
</table>

Table 5: Energy and Radio Metrics of All Architectures
<table>
<thead>
<tr>
<th>Deployment</th>
<th>Human Toxicity</th>
<th>Water Toxicity</th>
<th>Terrestrial Toxicity</th>
<th>Cobalt</th>
<th>Copper</th>
<th>Aluminum</th>
<th>Chemical Waste</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Configuration Deployment</td>
<td>353.41</td>
<td>399.86</td>
<td>404.27</td>
<td>428.18</td>
<td>364.49</td>
<td>805.17</td>
<td>1707.36</td>
</tr>
<tr>
<td>Low Durability Battery Deployment</td>
<td>418.81</td>
<td>473.86</td>
<td>479.08</td>
<td>507.43</td>
<td>431.94</td>
<td>954.18</td>
<td>2023.33</td>
</tr>
<tr>
<td>High Durability Battery Deployment</td>
<td>277.25</td>
<td>313.69</td>
<td>317.15</td>
<td>335.91</td>
<td>285.94</td>
<td>631.66</td>
<td>1339.43</td>
</tr>
<tr>
<td>Half Sensing Frequency Deployment</td>
<td>294.45</td>
<td>333.15</td>
<td>336.83</td>
<td>356.76</td>
<td>303.68</td>
<td>670.85</td>
<td>1422.54</td>
</tr>
<tr>
<td>Half Packet Rate Deployment</td>
<td>235.66</td>
<td>266.63</td>
<td>269.57</td>
<td>285.52</td>
<td>243.05</td>
<td>536.90</td>
<td>1138.50</td>
</tr>
<tr>
<td>No Timestamp Deployment</td>
<td>333.78</td>
<td>377.65</td>
<td>381.82</td>
<td>404.41</td>
<td>344.25</td>
<td>760.46</td>
<td>1612.55</td>
</tr>
<tr>
<td>Dense Coverage (9GWs)</td>
<td>289.47</td>
<td>327.51</td>
<td>331.12</td>
<td>350.71</td>
<td>298.54</td>
<td>659.49</td>
<td>1398.44</td>
</tr>
<tr>
<td>Extra Dense Coverage (13 GWs)</td>
<td>281.57</td>
<td>318.57</td>
<td>322.08</td>
<td>341.14</td>
<td>290.39</td>
<td>641.49</td>
<td>1360.28</td>
</tr>
<tr>
<td>Basic Configuration NBioT</td>
<td>267.56</td>
<td>302.72</td>
<td>306.06</td>
<td>324.17</td>
<td>275.94</td>
<td>609.57</td>
<td>1292.60</td>
</tr>
<tr>
<td>ILP Min Chemical Waste</td>
<td>344.53</td>
<td>389.81</td>
<td>394.11</td>
<td>417.42</td>
<td>355.33</td>
<td>784.93</td>
<td>1664.45</td>
</tr>
<tr>
<td>ILP Max QoS</td>
<td>354.03</td>
<td>400.57</td>
<td>404.98</td>
<td>428.94</td>
<td>365.13</td>
<td>806.59</td>
<td>1710.38</td>
</tr>
<tr>
<td>NBioT</td>
<td>267.56</td>
<td>302.72</td>
<td>306.06</td>
<td>324.17</td>
<td>275.94</td>
<td>609.57</td>
<td>1292.60</td>
</tr>
</tbody>
</table>

Table 6: Full Environmental Waste Metrics of All Architectures