Research by Experimentation
for Dependability on the
Internet of Things

D-4.4 – Final Integrated Prototype and Experiment

Grant Agreement no: 317826
www.relyonit.eu

Date: February 3, 2015

Author(s) and affiliation: Nicolas Tsiftes, Niclas Finne, Zhitao He, Thiemo Voigt (SICS), Faisal Aslam, Ioannis Protonotarios, Marco Zúñiga, Koen Langendoen (TUD), Carlo Alberto Boano, Felix Jonathan Oppermann, Kay Römer, Marcel Baunach (TUG), James Brown, Utz Roedig, Ibrahim Ethem Bagci, John Vidler (ULANC), Alejandro Veiga, Rafael Socorro (ACCIONA), Màrius Montón, José Carlos Pacho (WOS)

Work package/task: WP4
Document status: Final
Dissemination level: Public
Keywords: Internet of Things, sensor networks, interference, temperature

Abstract This document presents the outcome of Task 4.4—Final Integrated Experiment. The integrated experiment is carried out in the context of two business use cases: civil infrastructure monitoring and parking management. The integrated prototype is a refined version of that which we described and evaluated in D-4.3. We combine the components developed in WP 1-3, including newly designed and adapted protocols, protocol optimizations, environmental models, runtime assurance, and runtime adaptation. We evaluate the prototype in two realistic outdoor facilities made available by our industry partners, as well as in several testbeds. Whilst operating in challenging environments, the integrated prototype successfully attains key performance parameters as specified in the use case requirements. Lastly, an analysis of the results shows that the research conducted in RELYonIT is beneficial for the development of future business models by our industry partners.

Disclaimer

Contents

List of Figures

List of Tables

Executive Summary

In this deliverable, we present the results of our final integrated experiment, as part of Task 4.4—*Second Integrated Experiment and Prototype*. This experiment concerns a full integrated prototype of the components developed within work packages 1, 2, and 3. Earlier in the project, we conducted the first integrated experiment, which we documented in Deliverable 4.3.

In this task, we use the insights gained from the first integrated experiment to improve our integrated prototype. The first integrated prototype consisted of 1) learning of environmental models and parameters, 2) runtime assurance, and 3) newly developed and optimized protocols. The final integrated prototype extends this system by also including runtime adaptation, protocol parameter optimization, and a more comprehensive implementation of runtime assurance.

The integrated experiment is carried out in the context of two use case scenarios: civil infrastructure monitoring and outdoor parking management. Unlike the first integrated experiment, in which we focused on the outdoor parking management use case, we have added the civil infrastructure monitoring use case for the final integrated experiment. We use two realistic test facilities provided by our industrial partners ACCIONA and WOS. The DEMOPARK facility in Madrid, Spain is used to evaluate the integrated prototype for the building monitoring use case. We provide further experimental results from the TempLab testbed at TU Graz, which has been developed within RELYonIT to enable tests with repeatable temperature variations. The Smart City facility in Barcelona, Spain is used to evaluate the integrated prototype for the outdoor parking management use case. This part of the evaluation is focused on interference because of the facility's location centrally in the city, where there is an abundance of possible interference sources. Additionally, we stress test the integrated prototype in the FIRE testbed TWIST and a testbed at the University of Lancaster, where we can generate repeatable patterns of heavy interference.

Our results show that the components developed within the RELYonIT project successfully provide probabilistic bounds on the performance of protocols. In the final integrated experiment, we show that the integrated prototype for each application provides communication performance within the use case requirements. We have subjected the prototype to harsh environmental conditions, and shown that the RELYonIT system successfully upholds the performance within the required range. With the increased reliability, higher performance, and increased battery life that these results entail, our industry partners envision that the outcome of the RELYonIT project will have a positive impact on their future business models.

1 Introduction

This deliverable is a report on the outcome of Task 4.4—*Second Integrated Experiment and Prototype*. We present a prototype of a complete RELYonIT system, and provide an experimental evaluation of this prototype in a variety of testbed facilities both indoors and outdoors. The final integrated prototype is a refined version of the first integrated prototype, making use of the experience gained from Task 4.3—*First Integrated Experiment*.

In Deliverable 4.3 [18], we conducted the first integrated experiment in the context of Worldsensing's outdoor parking management use case (Smart Parking). For the final integrated experiment, we have considered ACCIONA's civil infrastructure monitoring use case in addition to the outdoor parking management use case. Hence, the final integrated experiment does not only test a more comprehensive RELYonIT system but also expands the scope of the experiments, covering one use case scenario from each industry partner.

The final integrated prototype consists of a variety of RELYonIT components that have been combined into a full system firmware. It encompasses newly developed and adapted protocols developed to provide probabilistic bounds for their performance under challenging environmental conditions. In this task, we have developed two prototype applications using these protocols: one for ACCIONA's civil infrastructure monitoring use case, and one for Worldsensing's smart parking use case. The selected protocols make use of environmental models that have been made for each use case scenario, and we have developed runtime assurance modules that execute in conjunction with the applications to ensure that the environmental models hold. In case environmental conditions surpass the bounds of the model, we use a runtime adaptation model to provide best-effort performance under such unusual conditions until the model is no longer violated.

The final integrated experiment tests the integrated prototype not only in the facilities provided by our industrial partners for each respective application but also in a variety of testbeds. These testbeds include the FIRE facility TWIST, which is hosted by TU-Berlin [12]; and the RELYonIT testbed TempLab, which is hosted by project partner TU Graz [5]. The facilities made available by ACCIONA and Worldsensing enable us to test our integrated prototype in a realistic setting for each application, whereas the testbed experiments enable us to test it in a large variety of temperature and interference conditions. Before describing each individual application and its corresponding experiment, we summarize their use case requirements in Chapter 2.

We divide the integrated experiment in two parts, studying the effects of temperature and interference independently. The reason for studying these effects independently in each deployments are 1) to isolate the environmental factors to properly be able to attribute the results in the experiments, and 2) to chose the dominating environmental factor at each testbed facility.

In Chapter 3, we report on the temperature part of the integrated experiment, which is carried out primarily in ACCIONA's DEMOPARK facility deployed outside the city of Madrid, Spain. Furthermore, we use the TempLab testbed deployed at TU Graz to further test different aspects of our protocols and models under a larger variety of temperature profiles in a controllable manner. At the DEMOPARK facility there are large temperature variations, whereas radio interference is low, as it is located in a remote area. Our results indicate that we can correctly predict and mitigate the impact of temperature variations on communication performance. Using the tools produced by RELYonIT we were able to sustain a packet reception ratio higher than 95% both in our testbed environment and in our target deployment in Madrid. Hence, using the tools developed within the RELYonIT project, we were able to meet not only the minimal performance requirements for our application scenario, but also the desired ones.

In Chapter 4, we report on the interference part of the integrated experiment, which is made primarily in Worldsensing's smart parking facility in the 22@ district in Barcelona, Spain. Furthermore, we extend our experiments by using the FIRE testbed TWIST and University of Lancaster's testbed. The smart parking facility exhibits lower the temperature variation because the nodes are placed in the tarmac, and are thus better shielded from the sun. Due to its central location in the city, however, we will experience a considerable degree of interference. Overall our results were positive and showed that we can capture radio interference at the deployment site through our environmental model. This was used to select appropriate parameters where we achieve the target listening duty cycle. Where the environmental differed, runtime assurance successfully detected these variations and runtime adaptation dynamically modified the system parameters to maintain targeted behaviour. Overall, we achieved a listening duty cycle of 0.74%, which is below the 1.38% target, and a total radio duty cycle of 1.42%.

In Chapter 5, we analyze the results of the final integrated experiment in relation to the use cases provided by the industry partners of the RELYonIT Consortium. Given the context of Chapter 5, we consider the implications of the results on future business models of the industry partners in Chapter 6. Lastly, we make our conclusions of the work carried out in this task in Chapter 7.

2 Use Cases and Requirements

The final integrated experiments implement prototypes of two use cases provided by the industry partners. The first use case, Smart Parking, was already chosen early in the project as documented in deliverable D-4.1. We summarize the properties of the use case below to make the document self-contained. The second use case, civil infrastructure monitoring was identified in D-4.1 as a relevant use case, but originally there was no plan to implement this use case. However, to support the exploitation of project results for industry partner ACCIONA, the consortium decided to implement civil infrastructure monitoring as a second use case, selected a demo site close to Madrid, and derived requirements for the specific instance of this use case. The use case and its requirements are described in this chapter, whereas the demo site and experiments are described in the subsequent chapter.

The two use cases are also chosen such that they are exposed to orthogonal environmental impact. The parking use case has interference as the dominating environmental influence due to its location in an urban area with lots of WiFi interference but where the sensor nodes are mostly shielded from direct sun radiation due to numerous buidlings and trees, therefore limiting temperature variations. In contrast, the civil infrastructure demo site is located in a rural area with little interference but with direct exposure to sunlight and resulting substantial temperature variations.

The two use cases also focus on two orthogonal dependability requirements. Due to the difficulty of replacing batteries in thousands of sensors embedded into tarmac, energy consumption is the most critical requirement for the smart parking use case. As the civil infrastructure monitoring use case may be applied in safety-critical contexts, packet delivery rate is the most critical requirement for the civil infrastructure monitoring use case.

Thereby we are able to *isolate* both the requirement of interest and the dominating environmental impact in each use case from other requirements and secondary environmental impacts, allowing use to clearly attribute performance observations to an environmental property.

2.1 Civil Infrastructure Monitoring Use Case and Requirements

In order to build energy-efficient buildings, it is very important to conduct careful tests on the new insulating materials to be used, to ensure that they reduce heat transfer sufficiently. Typically, short-term tests are carried out in specialized laboratories under a controlled environment with a stable parameter setting. The test results help designers estimate the energy efficiency of the considered material. Despite revealing extremely useful information during the early development stages, laboratory tests do not take into account the complex thermodynamics involved when a single construction element interacts with its surrounding elements and the environment over time. Therefore, these tests alone can not guarantee an insulation material's energy efficiency remains in line with the preliminary estimate after it is installed on a real building and used over a period of time. ACCIONA Infraestructures decides to compensate

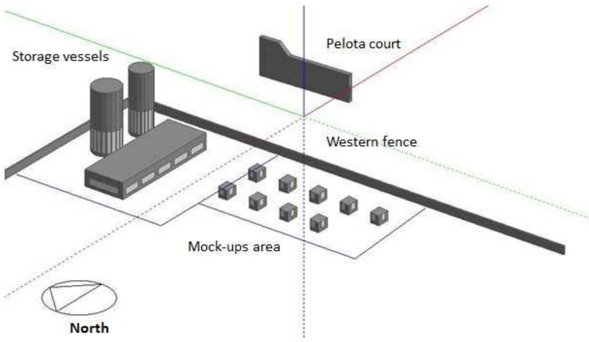


Figure 2.1: DEMOPARK facility mock-up.

this shortcoming by installing a real-world testing facility to compare the effectiveness of different building materials and HVAC systems. This facility mock-up provides extra insights that can be exploited when constructing new buildings (Figure 2.1).

Thermal tests for physical characterization of buildings are varied among projects. Commonly, they are focused on the study of temperature profiles and heat fluxes throughout façades, walls, roofs and other construction elements such as windows and doors, which creates a comprehensive map of the building's thermodynamic behaviour. Thereafter a statistical analysis over these variables is conducted; the results give rise to in-situ measurements of crucial physical parameters such as thermal resistivity and thermal conductivity, thermal inertia diagrams and averaged internal temperature and humidity profiles. Tests are complemented with HVAC systems working in summer or winter regimes, looking for differences in energy consumption and CO2 emissions.

For this reason, the monitoring system in charge of gathering all the measurements from the sensors should be as robust and reliable as possible. As these sensors may need to be added to an existing infrastructure and should remain operative after the completion of the building's construction, it is highly desirable that they are connected wirelessly. It has therefore been decided to use a wireless sensor network to connect these sensors together. Using small, battery-powered, wireless sensor nodes is also advisable as the sensor probes are usually located in hardly-accessible places (e.g., on the skyscraper's façade) where no wired infrastructure is available. For example, the system may be applied to buildings as the one shown in Fig. 2.2 in order to prevent conditions where ceramic tiles may fall off.

In order to allow precise studies on the insulation of a given material, it is fundamental that data loss is minimized, so that engineers have all the required information to draw conclusions about the effectiveness of material or HVAC system under study. Some of the tests to be carried out are heavily dependent on very small changes in the variables measured, so any gap in the collected data may lead to a false conclusion.

It is also fundamental to maximize the battery life of the sensor nodes to make sure that the network can remain operative for extended periods. Furthermore, as in the majority of cases sensor nodes will be deployed outdoors on the outside façade of buildings, it is vital for them to be able to cope with adverse environmental conditions (temperature, humidity, meteorological effects) in different climate areas (e.g., tropical, desert, temperate, or alpine) of the world.

Figure 2.2: Palau des Arts, Valencia (Spain). Ceramic material on the façade.

2.1.1 Characteristics of Interest

Critical QoS Requirements Data measurements are typically carried out periodically at timescales of tens of seconds, thus do not generate high traffic. In order to allow precise studies on the insulation of a given material, it is fundamental that packet data loss is minimized, so that engineers have all the required information to draw conclusions about the effectiveness of the studied material or HVAC system.

Impact on Quality of Service Sensor nodes will be placed outdoors, on the outside façade of the building, and they should hence be able to face adverse environmental conditions (temperature, humidity, meteorological effects) in different climate areas (e.g., tropical, desert, temperate, or alpine) of the world.

Different Phases of Activity in the Environment There are no activity changes expected at the deployment site.

Indoor / Outdoor Outdoor. Nodes will be installed on the outside façade of buildings.

Size (geographical and topological) The use case envisions networks with tens up to hundreds sensor nodes. An initial deployment will involve the deployment of seven nodes. Each node will control several sensors.

Scalability The system should be easily expandable: more sensors can be attached to a building and should be able to join the same wireless sensor network.

Repeatability As the sensor nodes are installed outdoors, we do not have any control on weather conditions and temperature fluctuations that may affect their performance negatively.

Reproducibility in testbeds A small-scale version of the use case environment can be reproduced in testbeds, provided that temperature fluctuations and/or the impact of weather conditions on sensor nodes and their communications can be reproduced.

New or existing deployment Existing deployment.

Access to Deployment Site The deployment site is owned by ACCIONA Infraestructures and it is located a few kilometres away from its R&D premises.

Node mobility All nodes at the deployment site will be static.

IP connectivity per individual sensors IP connectivity is not considered necessary.

Available radio spectra Radio interference could be present due to some construction machinery used at the deployment site. The sensor nodes employed will probably use the 2.4 GHz ISM band, and no particular interfering device in this frequency is expected.

2.1.2 Environmental Scenarios

The main environmental factors to be considered are temperature and humidity. We can identify the following different climate areas:

Cold climate

This climate is the one we can find, for example, in Russia or Austria. It is characterized by temperatures that can oscillate between −20 ℃ and up to 30 ℃, with a daily thermal oscillation of less than 10 ℃. The relative humidity in Moscow oscillates from 60% to 85%.

Temperate climate

This climate can be found in the Mediterranean (it is also known as cold Mediterranean). This climate can have thermal oscillation from −13 ℃ up to 45 ℃ during the year, and a daily oscillation of 10-15 ℃. We consider relative humidity to be similar to the previous one, in the range between 65% and 85%.

Warm desert climate

Saudi Arabia is an example of this. Here we can find temperatures from 18 ℃ up to 50 ℃. In these environments, we can usually find daily temperature oscillations of around 30 ℃. Humidity in Saudi Arabia goes from 11% of average in July to 46% in December, but sometimes humidity can reach 70%.

The expected environmental conditions for this use case are described in Table 2.1.

2.1.3 Dependability Requirements

ID	PM-1
Name	Data Loss $<$ 15%
Description	The system should not lose more than 15% of data packets.
Priority	M
Failure Effect	Not enough data for later studies.

Table 2.1: Environmental Conditions for Civil Infrastructure Monitoring.

Variable	**Min**	**Typ.**	**Max**	**Comment**
Temperature	-40 °C (We have observed down to -27 °C in Austria during winter).	0-12 °C Winter, 5-25 °C Autumn and Spring, 20-50 °C Summer time.	+85 °C (We have observed up to 70 °C in Qatar during summer under direct sun exposure).	Temperature range depends on the region of interest. Rapid temperature changes may occur if a sensor under direct sunlight exposure is covered. Note that temperature effects have day-night cycles.
Vibration&Shock	Sensors are not exposed to any major vibrations.	Sensors might be affected by wind or structure vibration.	In skyscrapers, sensors are heavily affected by structure vibrations and winds.	–
EM radiation	No EM radiation effects.	EM radiation form DC and AC sources such as power lines and building equipment.	Wi-Fi and radios may interfere with communications.	Affects detection performance and hence battery consumption and processing capabilities.
Humidity	No humidity, very dry environments. Might have effects on electronics.	45-70%, dry or wet normal operation.	Snow covering sensors or strong rain or flooding.	Strong effects of water on communications have been observed.

ID	PM-2
Name	Data Loss $< 5\%$
Description	The system should not lose more than 5% of the sensor measurements.
Priority	S
Failure Effect	Not enough measurement data.

ID	PM-3
Name	Battery life $>$ 2 months
Description	Battery needs to last for at least 2 months in an outdoor environment.
Priority	M
Failure Effect	System loses autonomy due to frequent manual battery replacements.

ID	PM-4
Name	Battery life $>$ 6 months
Description	Battery lasts for at least 6 months in an outdoor environment.
Priority	S
Failure Effect	System becomes uncompetitive against competing solutions.

2.2 Smart Parking Use Case and Requirements

Many European cities have been suffering a chronic shortage of parking space, especially in the city center. Still today, we can see many public spaces occupied by private vehicles as temporary parking lots. Cars driving around seeking free parking slots, known as hustle bustle traffic, is considered one of the most important contributing factors in traffic congestion in urban areas. Since parking is one of the fundamental services for inhabitants, commuters and visitors, parking management must be taken into account by traffic regulation authorities.

The architecture designed by Worldsensing in its smart parking product Fastprk is depicted in Figure 2.3. As described in D-4.1, the system is composed of a set of sensor installed in every parking spot and a set of gateways that collects the information and transmits the information to the central control using the Internet. There is also the option to install panels in the streets to inform the drivers about the number of available parknig places in real-time. The power requirements of the sytem are summarized in the Table 2.2.

Table 2.3 summarizes the environmental variables that can affect the smart parking scenario.

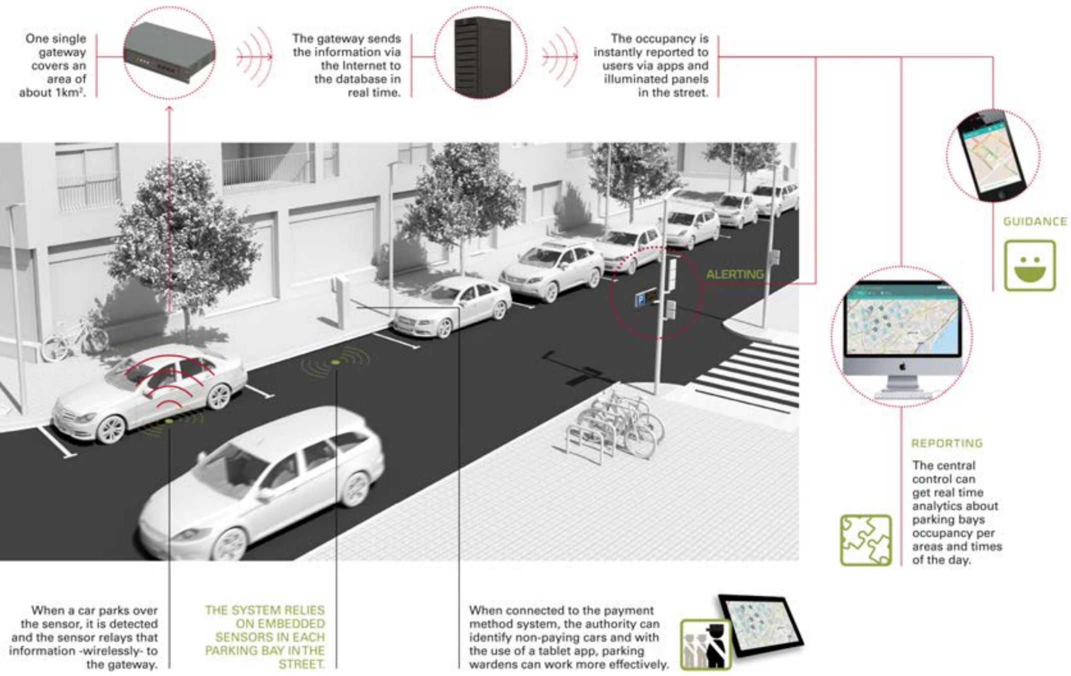


Figure 2.3: Smart Parking system diagram

2.2.1 Characteristics of Interest

Critical QoS requirements The most critical requirement in this case is the latency of the network. An event sent by a mote must arrive at the central server in time, within an upper bound of a few seconds.

Table 2.2: Power requirements for smart parking solution

Parking sensor component	Power Consumption	Operation time	Duty cycle
Radio transceiver	45 mW-150 mW	1-5 ms	0.1%-1%
μC	12 mW-150 mW	1-5 ms	0.1%-1%
Sensor	20 mW-30 mW	2ms	0.1%

Impact on Quality of Service It is expected that most of the environmental variations are going to be changes in temperature (depending on the direct exposure to the sun) and electromagnetic interferences (in urban areas). Added to this we must consider the fact that the node is embedded inside the road surface, so the radio signal has to be powerful enough to trespass the tarmac.

Different Phases of Activity in the Environment Changes in activity will follow a day-night cycle, because usually traffic movement is less frequent at night.

Indoor / Outdoor Outdoor.

Size (geographical and topological) Tens/Hundreds. We can consider aggregations of tens of motes covering house blocks.

Scalability For parking slots, the measures will be independent of each other. But multi-hop radio protocols usually limit the maximal diameter of the network (number of hops a packet can travel).

Repeatability Although cities in the world are located in very diverse climates, we assume that our three different scenarios cover the most typical climates. Therefore, it should not be too difficult to have a controlled environment like we presented in the sub-scenarios.

Control of environment We must consider, in addition to the typical conditions of any urban environment, that the motes are below parking slots. Communication is affected by the need for signal transmission through thin layers of concrete and cars, the presence of metal structures, and wet environments (oil, water). Nevertheless, we do not expect any severe problems in controlling these effects in our experiments.

Reproducibility in testbeds We do not perceive any difficulty in reproducing any of the environmental conditions in a lab environment.

New or existing deployment WOS has some deployments in different urban environments with different interference and climate to be used as testbed.

Access to Deployment Site The access to any mote is guaranteed, but usually a city permission is needed to cut streets. In most cases, we can be supported by city employees.

Node mobility We can assume everything is static.

Link to Internet There is no need for the nodes to have their own IP address.

Table 2.3: Environmental Conditions for Smart Parking.

Variable	Min	Typ.	Max	Comment
Temperature	-30 °C (We have observed down to -14 °C in Moscow during winter).	0-12 °C Winter, 5-25 °C Autumn, and Spring. 20-50 °C Summer time	+75 °C (We have observed up to +68 °C in south of Spain during summer under direct sun exposure).	-
Vibration&Shock	Streets are subject to constant vibrations centered at around 4-7 Hz, a few nm/s.	Cars passing and car motion generates vibrations up to 100 Hz and several mm/s.	Sensors may be subject to impact and shock from cars, load, and unload operations.	-
Magnetic radiation	Small towns and unpopulated areas show no magnetic disturbances.	Urban scenarios show DC magnetic disturbances mainly attributed to Metro and power stations.	In specific areas, very high magnetic noise will be present, including DC and AC components. AC are centered at 50Hz, 60Hz, and 100Hz.	It effects detection performance and hence battery consumption and processing capabilities.
Electromagnetic interferences	No interference due to very open space or no other devices	Some (<5) WiFi Access Points nearby, some occasional Bluetooth devices	Other WSN deployed nearby, lots of WiFi APs and Bluetooth devices (>10)	Interference can greatly affect in the performance of the entire network
Humidity	No humidity, very dry environments.	45-70%, dry or wet ground.	Snow covering sensors or rain.	We have observed strong effects of water on communications.
Car rotation	During night almost no movement. One car per 5 hours on average.	Normally in central zones we observe 3-4 cars per hour.	We have observed in very central areas, rotations are above 10 cars per hour.	It is also important to consider the traffic around sensors. At the center it can be very dynamic.

Available radio spectra As the electromagnetic spectrum is very crowded, it is necessary to select an available and suitable frequency for each specific location. The selection is usually limited to the unlicensed industrial, scientific and medical (ISM) radio bands as published by the ITU [13].

It is noteworthy that we detected radio interferences, which can become a main problem for this kind of applications. This is casued by the increasing congestion in the ISM band spectrum in cities, due to the use of WiFi and Bluetooth devices, other proprietory radio technologies and external factors such as diffuse electromagnetic noise sources (railway, power lines, microwave ovens or radio-links, etc.). If no change is made to a conventional radio solution, the application requirements will not be met in the near future.

We must either avoid these interferences, which is hard to do due to their nature, or try to modify our algorithms and protocols to co-exist with them. The second approach is taken in the Worldsensing use case.

2.2.2 Dependability Requirements

ID	SP-1
Name	Latency < 30 seconds
Description	The system has to have a response time below 30 seconds. Time response is considered as the time between a car change is detected by a mote and data is received by the Gateway.
Priority	M
Failure Effect	A misinformed driver might try to reach a slot only to find that it is already occupied, thus is forced to start a new search. This costs time and increases traffic, which in turn causes the driver to lose confidence in the system when the same system error recurs.

ID	SP-2
Name	Latency < 10 seconds
Description	The system should have a response time below 10 seconds. Time response is considered as the time between a car change is detected by a mote and data is received by the Gateway.
Priority	S
Failure Effect	A driver may be misled by the system that an vacant slot is available during busy hours.

ID	SP-3
Name	Data loss $< 10\%$
Description	The system does not lose more than 10% of the events
Priority	M
Failure Effect	System gives wrong information too often.

ID	SP-4
Name	Data loss $< 5\%$
Description	The system should not lose more than 5% of the events
Priority	S
Failure Effect	System gives wrong information sometimes.

ID	SP-5
Name	Data loss $< 1\%$
Description	The system could not lose more than 1% of the events
Priority	C
Failure Effect	System displays a reduction in quality of service.

The previous information allow us to establish the following requirements regarding the operational lifetime:

ID	SP-6
Name	Battery Life $>$ 6 months, Mediterranean city climate
Description	Battery has to last for at least 6 months in a climate characteristic of a Mediterranean city.
Priority	M
Failure Effect	System fails due to excessive maintenance costs.

ID	SP-7
Name	Battery Life $>$ 4 months, European continental city climate
Description	Battery has to last for at least 4 months in a climate characteristic of a European continental city.
Priority	M
Failure Effect	System fails due to excessive maintenance costs.

ID	SP-8
Name	Battery Life > 3 months, desert city climate
Description	Battery has to last for at least 3 months in a climate characteristic of a desert city.
Priority	M
Failure Effect	System fails due to excessive maintenance costs.

ID	SP-9
Name	Battery Life > 1 year, Mediterranean city climate
Description	Battery should last for at least 1 year in a climate characteristic of a Mediterranean city.
Priority	S
Failure Effect	System becomes uncompetitive due to increased maintenance costs.

ID	SP-10
Name	Battery Life > 8 months, European continental city climate
Description	Battery should last for at least 8 months in a climate characteristic of a European continental city.
Priority	S
Failure Effect	System becomes uncompetitive due to increased maintenance costs.

ID	SP-11
Name	Battery Life > 6 years, desert city climate
Description	Battery should last for at least 6 years in a climate as described for a desert city.
Priority	S
Failure Effect	System becomes uncompetitive due to increased maintenance costs.

ID	SP-12
Name	Battery Life > 2 years, Mediterranean city climate
Description	Battery may last for at least 2 years in a climate characteristic of a Mediterranean city.
Priority	C
Failure Effect	System becomes suboptimal.

ID	SP-13
Name	Battery Life $>$ 1.5 years, European continental city climate
Description	Battery may last for at least 1.5 years in a climate characteristic of a European continental city.
Priority	C
Failure Effect	System becomes suboptimal.

ID	SP-14
Name	Battery Life $>$ 1 year, desert city climate
Description	Battery may last for at least 1 year in a climate characteristic of a desert city.
Priority	C
Failure Effect	System becomes suboptimal.

3 Madrid Deployment: Temperature

This section describes the final integrated experiment focusing on the impact of temperature variations on IoT performance carried out within a practical and realistic application scenario provided by the ACCIONA industry partner.

Section 3.1 describes the setup of our experiments and details on the facilities where the deployment has been carried out and on the installation of the wireless sensor nodes. Section 3.2 shows how the performance requirements cannot be met when using state-of-the-art communication protocols due to the high temperature variations at the deployment location. We further show that by running the same software on TempLab, our augmented testbed infrastructure with the ability to replay temperature profiles developed within WP4, we obtain the same network performance as in the real-world. This indicates that our testbed infrastructure can be used to faithfully replicate the impact of temperature on IoT hardware and protocols and to investigate their impact on communication performance.

We then apply the solutions developed within RELYonIT and show that they allow to meet the specified performance requirements, both in our TempLab testbed environment, as well as at the target deployment. In particular, we describe in Section 3.3 how we apply the RELYonIT framework to mitigate the impact of temperature on IoT performance, integrating the results of WP1, WP2 and WP3. We then show in Section 3.4 the experimental results obtained running RELYonIT solutions on controlled settings as well as at the target deployment. Using the tools produced by RELYonIT we were able to meet not only the minimal performance requirements for our application scenario, but also the desired ones.

3.1 Experimental Setup

In order to carefully test how the insulating materials used in the construction of buildings reduce heat transfer, ACCIONA Infraestructures installed a testing facility to compare the effectiveness of different building materials and Heating, Ventilating, and Air Conditioning (HVAC) systems in a real-world setting.

Fuente del Fresno facility. The facility, called DEMOPARK, has a total area of 1200 m^2 and is located in Fuente del Fresno, 20 km North of the city of Madrid, Spain, as shown in Figure 3.1[1].

[1] The facility has been installed in the context of the European consortium NANO E2B CLUSTER composed of six research projects funded under the working topic "New nanotechnology-based high performance insulation systems for energy efficiency". These projects include 11 large enterprises, 16 small and medium enterprises, 17 research and technological organizations and 5 universities. DEMOPARK is an example of European cooperation in Research and Development issues, and symbolizes the firm compromise of its members to reach the 20-20-20 Objectives using 100% European cutting-edge technologies. The facility will be used in the near future by additional European projects to test their insulation elements on site.

Figure 3.1: Location of the ACCIONA facility used to test the effectiveness of insulating materials. Aerial images taken from `sigpac.mapa.es`.

Figure 3.2: Overview of the ACCIONA facility with the buildings used to test insulating materials. Aerial images taken from `sigpac.mapa.es`.

The climate in Madrid's area is typically Mediterranean, and the air temperature outdoors can vary by up to 50 °C across one year. This facility is therefore a perfect fit to test the solutions produced by the RELYonIT consortium with respect to the impact of temperature variations on IoT performance.

Figure 3.2 illustrates the DEMOPARK facility in detail. The facility contains several testing rooms to test assorted materials such as coatings reflecting the infra-red part of the solar spectrum, phase change materials integrated in lightweight insulation panels, as well as lightweight vacuum panels with ultra-low thermal conductivity. In Figure 3.2 one can recognize 12 squared buildings of approximately 2.5 meters width and 2.8 meters height: each of them is a testing

(a) (b)

Figure 3.3: Examples of the materials used to upholster the inner room of the buildings shown in Figure 3.2.

room designed and built carefully following researchers guidelines. Each testing room counts on its own monitoring system and allows to cope with any sensor distribution. The cores of the monitoring system are currently data loggers, but ACCIONA is planning to use a wireless sensor networks measuring physical parameters periodically and access them via the Internet in real-time from any location world-wide.

Figure 3.3 illustrates an example of the special materials installed inside the testing rooms. Thermal tests typically focus on the study of the temperature profiles and the heat fluxes throughout façades, walls, roofs, and other constructive elements such as windows and doors in order to create a comprehensive map of the building's thermodynamic behaviour. Thereafter, a statistical analysis over these variables is conducted, and the results are used to drive in-situ measurements of crucial physical parameters such as thermal resistivity, thermal conductivity, thermal inertia, and averaged internal temperature and humidity. Tests are complemented with HVACs systems working in summer or winter regimes, looking for differences in the energy consumption and CO_2 emissions. Finally, the facility can also count on its own weather station, with an independent pyranometer to measure global irradiance on site over the year.

Installation. We install 7 Maxfor CM5000 and Moteiv TelosB wireless sensor nodes (Tmote Sky replicas) on the northern building in the facility, as shown in Figure 3.4. The sensor nodes are installed on the different outdoor façades of the building and their exact position is summarized in Table 3.1. Top and bottom indicate whether the sensor node is placed at approximately 2.5 or 0.5 meters height, respectively.

All nodes are USB-powered[2] in order to allow a real-time gathering of the sensor data and communication statistics. This allows us to have a bird-eye view on the data collection and

[2]Please note that the USB connection is not used for the actual communication, but only to collect the statistics about communication performance.

Node ID	Façade	Height
101	South	Top
102	South	Bottom
103	East	Top
104	West	Bottom
105	North	Top
106	North	Bottom
107	West	Top

Table 3.1: Position of the wireless sensor nodes within the outdoor façades of the building. Top and bottom indicate whether the sensor node is placed at approximately 2.5 or 0.5 meters height, respectively (see Figure 3.4).

Figure 3.4: The wireless sensor nodes have been installed on the Northernmost building in the facility. A Webcam pointed on the south-west façade of the building has also been installed to monitor and record weather conditions.

on the state of communication protocols, and to verify whether the performance requirements, i.e., a minimum packet reception rate (PRR) of 85% and a desired one of 95%, have been met at any point in time. All communication statistics are logged through Contiki's `serialdump` software on a central computer with access to the Internet. This allows us not only to have real-time access to the collected data, but also to easily reprogram all nodes remotely.

(a) (b) (c) (d)

Figure 3.5: Images from the Webcam at different times of the day, namely dawn (a), midday (b), sunset (c), and late evening (d).

(a) (b)

Figure 3.6: Installation of the sensor nodes on the different outdoor façades of the building during October 2014.

To monitor and record weather conditions and verify anomalies in the experimental setup, we place an Axis 214 Webcam pointed on the south-west façade of the building. Images are recorded every minute and transferred to a remote FTP server. This allows us to monitor the activities in the building and the sun exposure 24/7 (see Figure 3.5 for an example) and to verify whether our experimental results and sensor measurements are consistent.

The installation of the wireless sensors nodes took place in the middle of October 2014 and the setup (including the Webcam) was fully operative on October 28 (Figure 3.6). The seven nodes form a star topology, with node 101 (South façade) being designated as sink. Nodes are enclosed into IP44 enclosures[3] and use their embedded PCB antenna for their communications (no external SMA antenna was used in our experiments). Because of the low-gain antenna,

[3] We have used enclosures of different colors (white, gray, black) in our experiments. Although we would have expected the nodes packaged in the black enclosures to suffer a far higher on-board temperature variation compared to the ones packaged in brighter enclosures, this was not the case throughout our experiments. Darker enclosures caused temperature to raise only by a few Celsius degrees.

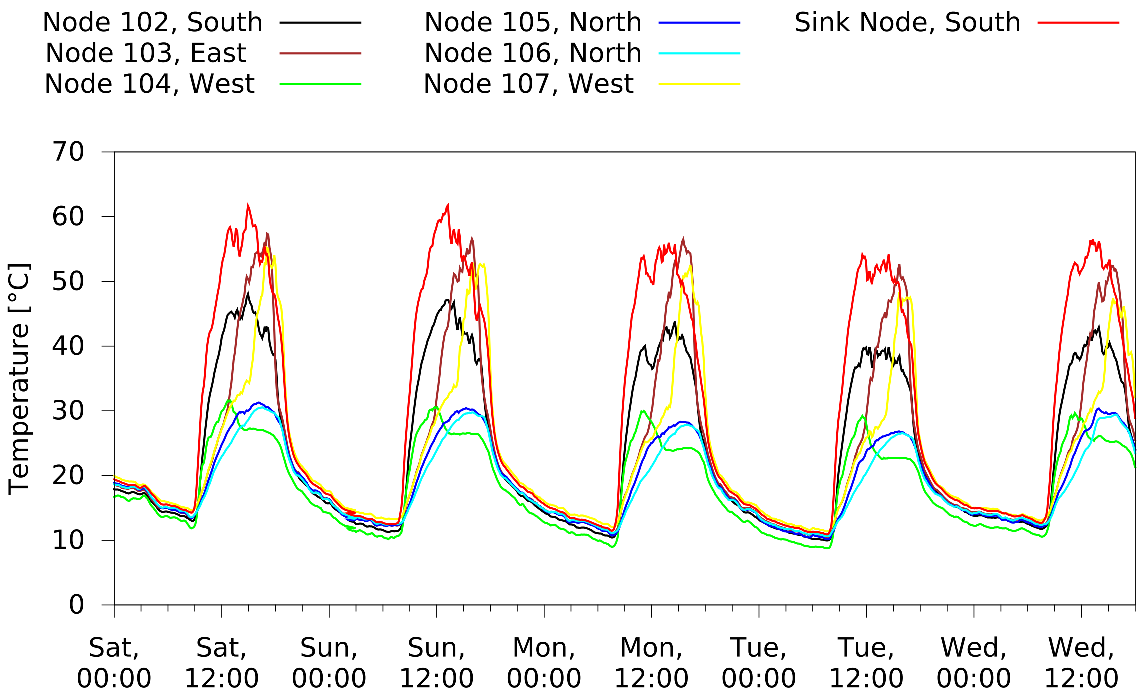


Figure 3.7: On-board temperature of the wireless sensor nodes recorded during the end of October. Nodes deployed on the South façade experience the highest temperature fluctuations. Each node samples temperature every 10 seconds.

nodes 103 and 104 were at the border of their communication range also when using their highest transmission power, with limited packet reception rate. The transmission power of the leaf nodes has been selected in such a way that the sink can be reached within one hop (if possible). The sink, instead, uses always its maximum transmission power (0 dBm) to acknowledge all incoming messages: this allows us to avoid asymmetric links.

Setup validation. We validate the experimental setup during the last week of October as follows. All sensor nodes run the Contiki operating system [11]: each node runs a custom software measuring the on-board temperature every 10 seconds and logging this information on the central computer[4]. Figure 3.7 shows the on-board temperature of the wireless sensor nodes over 120 hours. Daily temperature fluctuations are clearly visible for all nodes deployed in the facility. In particular, nodes deployed on the South façade experienced the highest temperature fluctuations, with temperature values between 8 and 62 °C across seven days. Nodes deployed on the North façade experienced the smallest temperature fluctuations, with daily fluctuations up to 25 °C. The fluctuations of on-board temperatures have been persistent throughout the

[4]This software consists mostly of the environmental model tool described in Section 3.3. The software can be found in the attached zip file in the folder `D-4.4/temperature/environment_collection`.

duration of our experiments (running uninterruptedly from the end of October to the end of January). The maximum on-board temperatures recorded during December were approximately 10 °C lower than the ones measured during October, nevertheless the daily fluctuation remained consistent. For example, the sink node (ID 101) regularly suffers a fluctuation of about 55 °C between day and night during sunny days: during October temperature varied between 5 and 60 °C; whilst during December (the coldest time-frame), the recorded temperature were typically between -5 and 50 °C.

We further verify the absence of radio interference in our experimental setup, to avoid mixing up the effects of different causes for the quantitative evaluation. We collect RSSI measurements in absence of packet transmissions and observe that the noise floor remains pretty stable below -90 dBm without visible bursts of interference throughout the day. The facility is indeed located in a remote area outside the city of Madrid and the amount of radio interference in the surroundings is minimal. 3G connection is used to remotely connect to the central gateway computer, which does not introduce coexistence problems in the 2.4 GHz ISM frequency band used by the Maxfor CM5000 and Moteiv TelosB wireless sensor nodes.

3.2 Data Collection using State-of-the-Art Communication Protocols

We use the experimental setup described in Section 3.1 to carry out a data collection sampling the properties of insulating materials resembling the one intended by ACCIONA for their civil infrastructure monitoring system[5]. In this initial phase, we employ state-of-the-art communication protocols without resorting to the techniques developed within RELYonIT, showing that they are not sufficient to achieve the application requirements for this specific use-case.

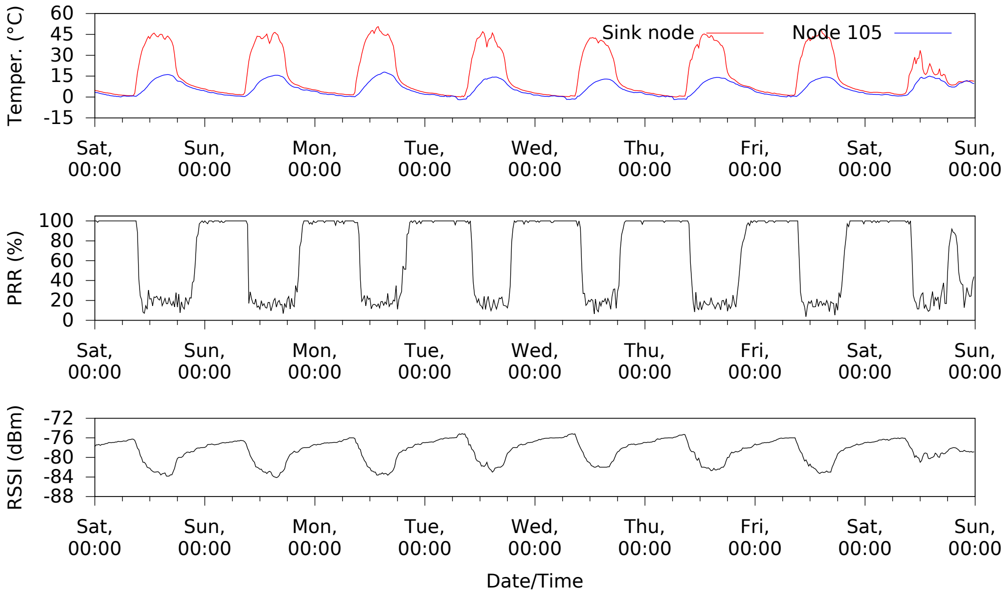
Network architecture. Our data collection collects statistics about the reception of packets together with the time-stamp for each transmission and reception, as well as the on-board temperature of the sensor nodes. At the physical layer and at the link layer, we use standard IEEE 802.15.4 communication. We employ ContikiMAC [8] as radio duty cycling protocol and Carrier Sense Multiple Access (CSMA) as MAC protocol. At the network layer and above, we use the Rime communication stack [10] and a custom data collection in which all sensor nodes send periodic reports to the sink and obtain an acknowledgment about their correct reception.

All nodes run the default ContikiMAC with a fixed CCA threshold (CCA_{Thr}) of -77 dBm.

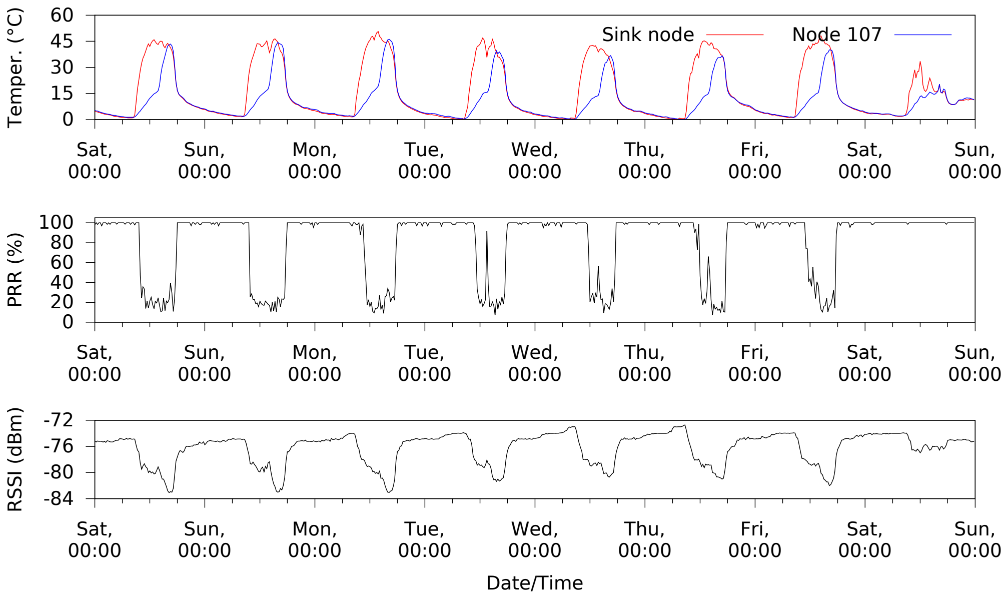
3.2.1 Insufficient Performance at Target Deployment

As we have shown in Figure 3.7, daily fluctuations of on-board temperature can be as high as 55 °C for some of the sensor nodes in our deployment. We have shown in [22] and [23] that such temperature variations can cause an attenuation of the received signal strength at the sink in the order of several dB – a variation sufficient to push a stable link into the transitional or disconnected region and to cause significant problems at common CSMA-based data link layer protocols such as ContikiMAC [3–5].

[5]The code can be found in the zip file on the folder `D-4.4/temperature/data_collection`.



(a) Link between node 101 and 105



(b) Link between node 101 and 107

Figure 3.8: Performance of default's ContikiMAC ($CCA_{Thr} = -77$ dBm) in our deployment at the DEMOPARK facility in Madrid. The experiment took place at the end of December 2014 and clearly shows that the performance offered by state-of-the-art tools is insufficient for our purposes.

Node ID	Average PRR	Minimum PRR
102	91.14%	42.42%
105	59.73%	3.63%
106	19.90%	3.77%
107	74.70%	7.04%
Network	61.38%	–

Table 3.2: Performance of default's ContikiMAC ($CCA_{Thr} = -77$ dBm) in our deployment at the DEMOPARK facility. The experiment took place at the end of December 2014.

Our experiments indeed confirm that the communication performance varies dramatically between day and night, and that several nodes are unable to meet their performance goals (a packet reception of at least 85%, better 95%). These results resemble the ones presented by Wennerström et al. [20] in their long-term outdoor deployment in Uppsala, Sweden, where they experienced significantly higher performance during winter and during night-time.

Figure 3.8 shows that some of the links between the sink and the leaf nodes have suffered a large decrease in performance during a long-term experiment carried out between the end of December and the beginning of January. Indeed, the link between node 101 and 105 (Figure 3.8(a)) and 101 and 107 (Figure 3.8(b)) are completely compromised during daytime: the PRR of both links decreases down to 0% during daytime as a consequence of a reduction of almost 10 dB in the received signal strength. This attenuation of the received signal strength causes an intersection with the fixed CCA threshold at high temperatures: as we have pointed out in [4], this compromises the wake-up mechanism at the receiver node and renders the link useless.

The performance of each individual link is summarized in Table 3.2. Please notice that the links between the sink and nodes 103 and 104 are not included as these nodes cannot achieve a reliable performance even when communicating at the highest transmission power. As we will show in Section 3.4, our parametrization tool developed in WP3 is able to detect the unreliability of these links and automatically warns the user about the problem.

Overall, our network can correctly deliver only 61.38% of the packets in total, with a highly different performance between daytime and night-time. This performance is not sufficient to satisfy the requirements of this application scenario (i.e., the delivery of at least 85% of the packets to allow a precise study of the quality of the insulating material) and we therefore need to resort to more dependable solutions to bring the use case scenario to reality.

3.2.2 Replication of Environmental Impact on Augmented Testbeds

We study more accurately the impact of the temperature variations found in our deployment scenario using one of the testbed infrastructures augmented with the ability of reproducing realistic environmental effects. These testbed extensions, developed in WP4, play a crucial role for the investigation of protocol performance, as they allow to rerun experiments under almost identical environmental conditions.

Node ID	Minimum temp. [°C]	Maximum temp. [°C]	Temp. difference [°C]
101	10.88	62.29	51.41
102	9.96	48.10	38.14
103	10.50	57.66	47.16
104	8.74	31.81	23.07
105	10.14	31.31	21.17
106	10.56	30.57	20.01
107	11.37	55.78	44.41

Table 3.3: Temperature variations recorded by the sensor nodes deployed at the DEMOPARK facility in Madrid during the end of October 2014.

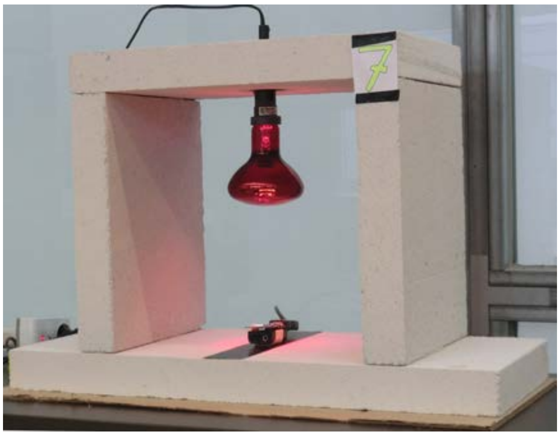
TempLab facility. TempLab is an extension for WSN testbeds that allows to control the on-board temperature of sensor nodes and to study the effects of temperature variations on network performance in a precise and repeatable fashion. TempLab can accurately reproduce traces recorded in outdoor environments with fine granularity, while minimizing the hardware costs and the configuration overhead. TempLab can be used to analyse the detrimental effects of temperature variations on processing performance, as well as on routing and data link layer protocols, deriving insights that would have not been revealed using existing testbed installations [2, 5].

We use our TempLab facility deployed at Graz University of Technology to replay the same temperature patterns recorded in Madrid. Figure 3.9 shows the setup of the testbed infrastructure. We form a network with star topology that resembles the one deployed at the DEMOPARK facility in Madrid, using node 200 as a sink and nodes 202, 203, 205, 208, 209, and 213 as leaf nodes. The transmission power of the nodes was selected in such a way to obtain a received signal strength similar to the one between the links in our original DEMOPARK deployment.

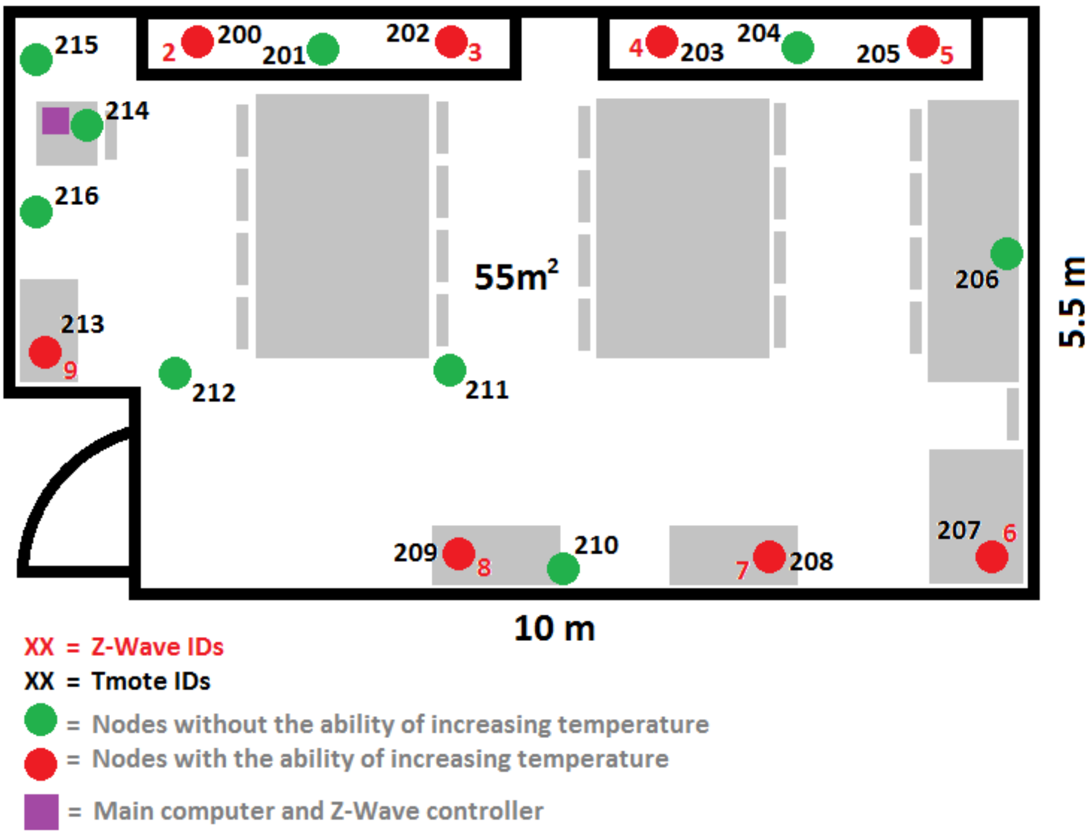
Replication of temperature. We run the same data collection software described in Section 3.2 and playback the temperature patterns recorded at the DEMOPARK facility in Madrid during the end of October (Figure 3.7). Because our testbed facility has available mostly heating-only nodes that do not provide cooling capabilities, we have increased the temperature values shown in Table 3.3 by approximately 17 Celsius degrees, so that the minimum temperature was above ambient temperature and could therefore be reached when the infra-red heating lamps are cooling down. The source code of the TempLab files can be found in the zip file on the folder `D-4.4/temperature/templab`.

Our experimental results show that the performance of the sensor nodes heated in the testbed exhibit the same decrease observed in the DEMOPARK facility in Madrid. Figure 3.10 shows for example two of the heated links in the testbed: the packet reception rate drastically decreases at high temperatures, resembling the results observed in Figure 3.8. The link between node 200 and 203 (Figure 3.10(a)) sustains indeed only 68.36% PRR, whereas the link between node 200 and 213 exhibits a PRR of 72.19% (Figure 3.10(b)).

(a)

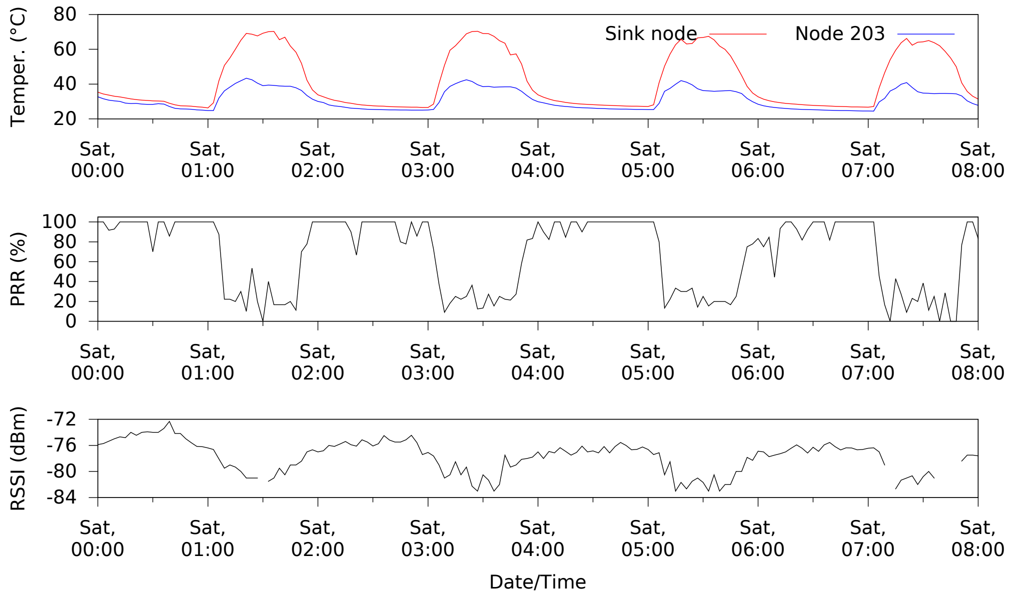


(b)

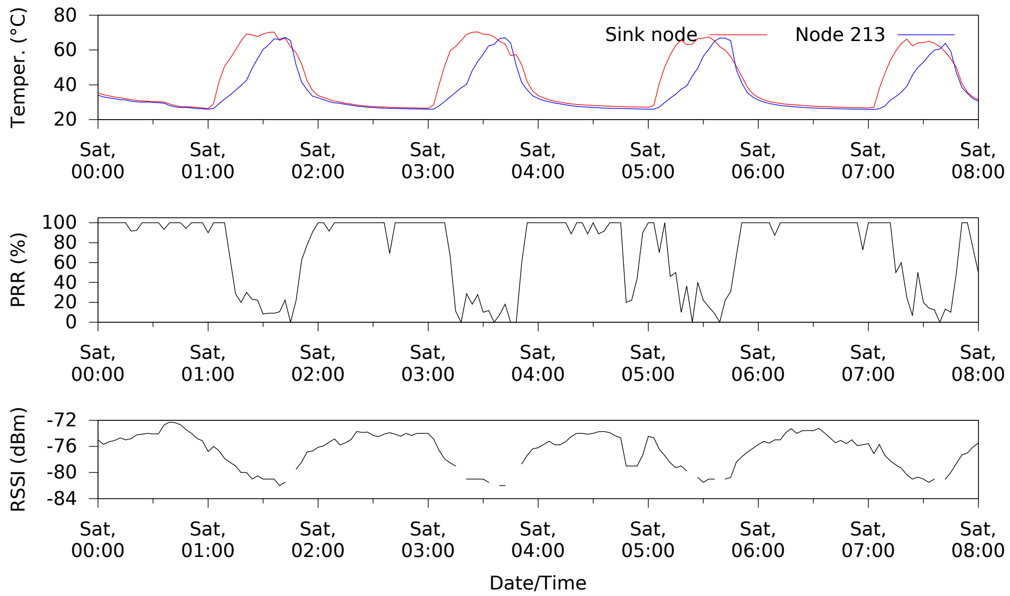


(c)

Figure 3.9: Overview of the TempLab testbed infrastructure in Graz (a) with infra-red heating lamps on top of each sensor node to control their on-board temperature (b) [4]. The node IDs of the sensor nodes with heating capabilities are depicted in red (c).



(a) Link between node 200 and 203



(b) Link between node 200 and 213

Figure 3.10: Performance of default's ContikiMAC ($CCA_{Thr} = -77$ dBm) in our indoor testbed replaying the temperature profiles encountered in our target application scenario.

3.3 Prototype Description

To achieve the desired performance, we use the techniques developed within WP1, WP2, and WP3 of RELYonIT. We use the temperature information available from the DEMOPARK facility (Table 3.3) to derive the environmental model capturing the maximum temperature variation on each of the nodes. We further use the platform model described in [3, 22] to approximate the signal strength attenuation at high temperatures and infer this information to the static protocol parametrization tool developed in WP3. The latter derives an optimal initial configuration of the data link layer protocols used in our experiments, namely:

1. TempMAC, the adaptive data link layer protocol developed within WP2 [4] adjusting its clear channel assessment threshold CCA_{Thr} to temperature changes. TempMAC embeds runtime adaptation as it automatically learns the temperature changes in the surrounding environment and adapts the CCA threshold accordingly (even if temperature has exceeded the expected model bounds);

2. ContikiMAC, the default radio duty cycling protocol of Contiki that uses a fixed CCA threshold. Although TempMAC is superior to ContikiMAC, we employ the static protocol parametrization tool developed in WP3 to derive an optimal initial configuration for ContikiMAC's CCA_{Thr} and show that runtime assurance and adaptation is required when using a fixed CCA threshold. Indeed, in the presence of temperature variations that exceed the expected model bounds, ContikiMAC cannot adapt to the new environment and a new CCA_{Thr} needs to be explicitly computed.

In Section 3.4 we will run the data collection using TempMAC and ContikiMAC with the optimized CCA_{Thr} on both our TempLab facility in Graz and on the DEMOPARK facility in Madrid, demonstrating that RELYonIT solutions can indeed satisfy the desired performance requirements and that they constitute a significant improvement with respect to the state-of-the-art.

Environmental model tool. For this final integrated prototype, we used an improved version of the environmental model data collection tool presented in D-1.2 [6] and D-4.3 [18]. The source code for the environmental model data collection tool can be found in the `D-4.4/temperature/environment_collection` directory of the attached zip file. The tool is executed on each of the nodes at the deployment site to collect the temperature profiles 24/7 and to create instances of the environmental model as described in Section 3.1. The tool measures temperature every 10 seconds using the on-board SHT11 sensors. This data is then summarized and used by our WP3 configuration tool in order to determine appropriate parameter selection for the data link layer protocols used in our experiments.

Platform model. We exploit the platform model used in the first integrated experiment [18]. This model allows us to predict the attenuation or strengthening of the wireless signal in the presence of a temperature variation for a given hardware platform and states that $s_r' = s_r - \Delta T_{tx}\alpha - \Delta T_{rx}\beta$. s_r' represents the predicted RSSI at the worse-case temperature T_{worse}, s_r is the original RSSI value measured at a given temperature $T_{initial}$, whereas T_{tx} and T_{rx} represent the difference between T_{worse} and $T_{initial}$ at the sender and receiver, respectively.

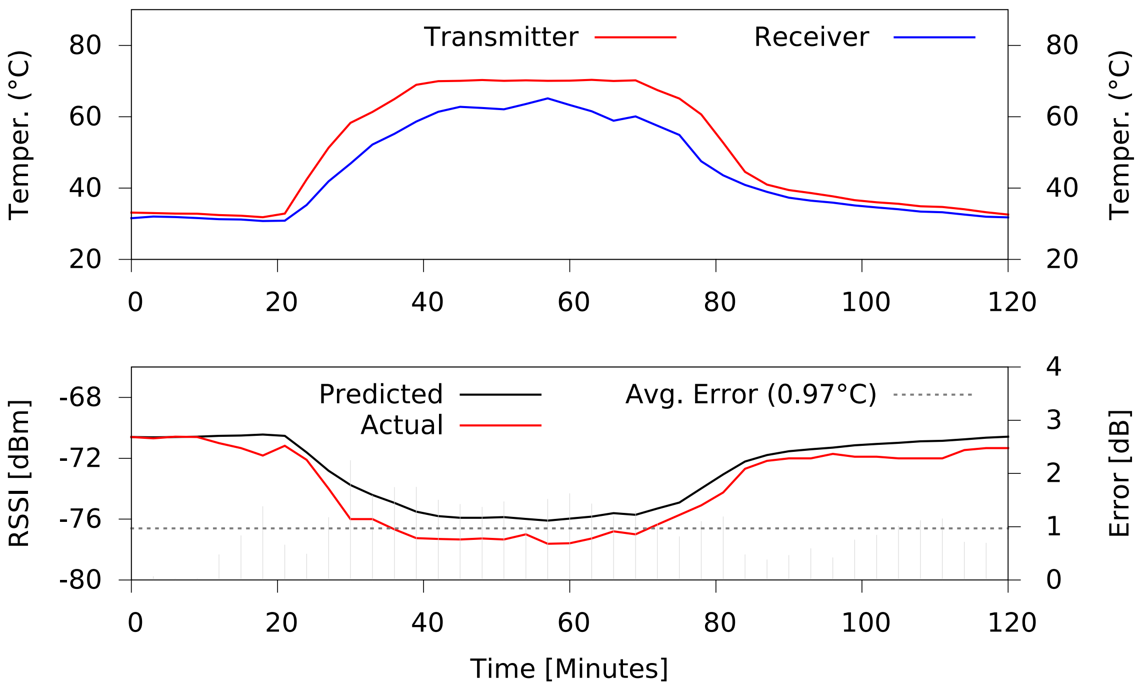


Figure 3.11: Validation of the platform model used in our experiments.

α and β represent respectively the attenuation of the signal on the transmitter and on the receiver side of a given radio transceiver, and have been computed following the approach described in [3]. In our experiments we use $\alpha = \beta = 0.078$.

We validate how good these values capture the characteristics of the platform in use by comparing the actual attenuation of signal strength with the one predicted by the platform model. We consider the link between node 200 and 201 and pick an RSSI value at relatively low temperature. We then use TempLab to vary the temperature of the node and record the temperature variation and the RSSI of the link, computing the difference between the latter and the one computed using our platform model. Figure 3.11 shows the attenuation of the wireless signal at high temperature together with the difference between the predicted and the actual RSSI. This difference is on average 0.97 dB: given that RSSI measurements are integer values, we can consider such error acceptable, as it is below the unit.

Optimized and newly designed data link layer protocols. Most CSMA-based protocols such as ContikiMAC use a fixed CCA threshold CCA_{Thr} that does not change at runtime. We have pointed out in [4] that the selection of the initial CCA_{Thr} is fundamental, as at high temperatures the signal strength may attenuate and cause the destruction of the wireless link if it reaches values below CCA_{Thr}. As an output of our work in WP3, we have developed a static protocol parametrization tool that can use the environmental and platform models derived in WP1 to compute an optimal CCA_{Thr} that guarantees a certain performance (see "selection of the initial CCA threshold" section below).

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dependabilityReq xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:noNamespaceSchemaLocation="schema.xsd">
  <name>TempMAC Demo Requirements</name>
  <protocol_class>tempmac</protocol_class>
  <objective>CCA</objective>
  <criteria>MAX</criteria>
  <constraints>
    <constraint>
      <metric>YIELD</metric>
      <threshold>0.85</threshold>
      <probability>1.0</probability>
    </constraint>
  </constraints>
</dependabilityReq>
```

Listing 3.1: Requirement specification for the Madrid demo scenario.

The latter can provide an initial CCA threshold for ContikiMAC that is guaranteed not to intersect the received signal strength at high temperatures, ensuring correct operations despite the use of a fixed CCA threshold.

To mitigate the impact of temperature variations on CSMA-based protocols at runtime, we have further developed TempMAC, an extension for existing data link layer protocols that dynamically adapts the clear channel assessment threshold to temperature changes. It does so based on the temperature measured locally and on the highest temperature measured across all neighbouring nodes [4]. The information about the highest neighbour temperature is traditionally inferred from the routing layer. As in our application scenario we have a star topology, we embed TempMAC in our data collection application and piggyback the temperature information of each node in the packets. Each node locally stores the highest temperature from any incoming neighbour and uses this information to compute ΔT_{tx}. Also in this case, to make sure that we can guarantee the desired performance, we use the static protocol parametrization tool developed in WP3 to compute an optimal initial CCA threshold CCA_{Thr}, i.e., the maximum CCA_{Thr} that satisfies a certain performance requirement. Indeed, the higher CCA_{Thr}, the lower the false wake-up rate (and therefore the energy consumption of the nodes), as pointed out by Sha et al. [16]. The source code for the data collection running with either ContikiMAC or TempMAC can be found in the `D-4.4/temperature/data_collection` directory of the attached zip file. The radio duty cycling protocol used in the experiments is specified in the `project-conf.h` file.

Selection of the initial CCA threshold. In order to select an optimal configuration of the initial CCA threshold of data link layer protocols, we employ the static protocol parametrization tool developed in WP3. The latter employs an implementation of the corresponding protocol model specified in WP2 [24]. In particular, the tool uses the model shipped with the parametrization framework (that can be found in the folder `D-4.4/temperature/parametrization_tool/models/tempmac`) and integrates a suitable configurable temperature and platform model

```
#include "relyonit-configuration.int.h"

const struct relyonit_configuration relyonit_configuration = {
  1,
  (struct relyonit_performance_state []) {
    {
      "TempMAC_Demo_Requirements",
      1,
      (struct protocol []) {
        {
          "tempmac",
          1,
          (struct parameter []) {
            {
              "cca",
              INT,
              -77,
            },
          },
        },
      },
    },
};
```

Listing 3.2: Resulting configuration for the Madrid demo scenario.

as developed in WP1. In addition, the static parametrization tool uses as input the connectivity of all sensor nodes in the target deployment. The latter is used to compute the packet reception rate of individual links.

The parametrization tool itself consists of a Python application that employs mathematical optimization to derive near-optimal parameter values to configure a specific protocol based on a previously characterized environment. The entry point to the application is the Python script that can be found in the attached zip file on the `D-4.4/temperature/parametrization_tool/main.py` folder. Once run to completion, the Python script generates a C file that contains the required protocol parameters. A detailed description of the software and the intended work flow can be found in deliverable D-3.2 [15].

In our specific demo setup, we collect the following inputs: (1) signal strength readings for each possible link at different sending powers and the corresponding temperature range at which these readings were collected (this is the output of the tool computing the connectivity in the deployment or testbed environment and can be found in the zip file on the `D-4.4/temperature/connectivity_tool` folder); (2) noise readings for each node and the corresponding temperature range at which these readings were collected; and (3) encountered temperature variations during the pre-deployment phase, which allows to determine the expected temperature range on which the models are parametrized. These traces are finally read by the data link layer model and are used to configure the specific model instance that represents either the TU Graz TempLab testbed or the DEMOPARK facility in Madrid. For

Node ID	Average PRR	Minimum PRR
202	97.86%	75.61%
203	98.44%	84.85%
205	97.07%	66.67%
208	98.78%	87.50%
209	96.37%	44.83%
213	96.09%	16.67%
Network	97.43%	–

Table 3.4: Performance of TempMAC (CCA_{Thr} = −76 dBm) in our TempLab testbed when replaying the same temperature profiles recorded in Madrid. Using RELYonIT solutions, we can meet the specified performance goals (PRR ≥ 95%).

both networks, we employ a requirement specification (see Listing 3.1) based on the scenario requirements (PRR ≥ 85% or 95%). The process yields a C file with the optimal configuration for the data link layer protocol, i.e., either TempMAC or ContikiMAC (see Listing 3.2).

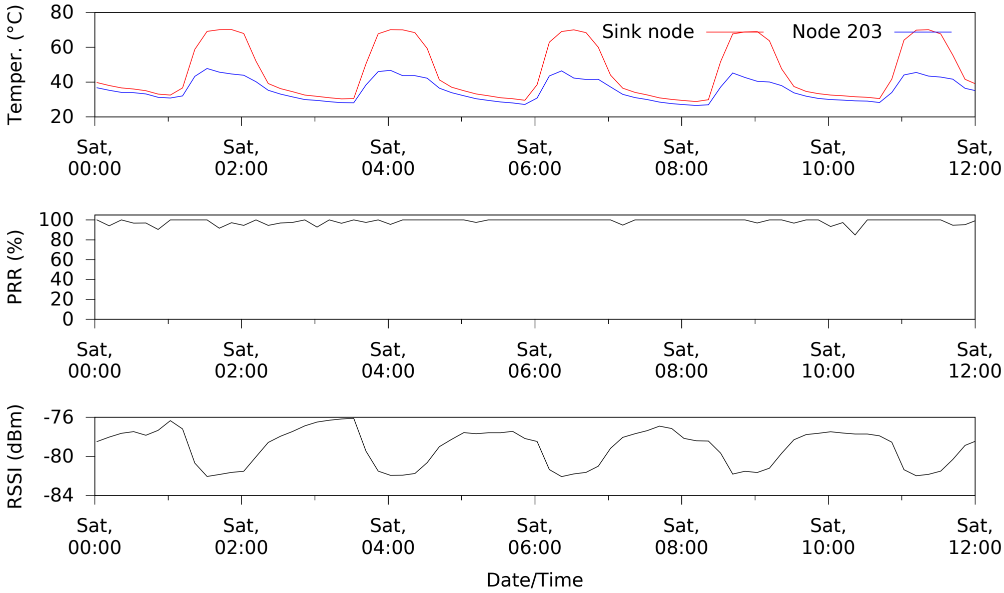
3.4 Results

We now apply the solutions developed within RELYonIT described in Section 3.3 and show that they allow to meet the specified performance requirements, both in our testbed environment (Section 3.4.1), as well as at the DEMOPARK facility in Madrid (Section 3.4.2). We further evaluate the performance of our runtime assurance and adaptation tools in Section 3.4.3, comparing the performance of ContikiMAC using different model inputs and highlighting their correct operation.

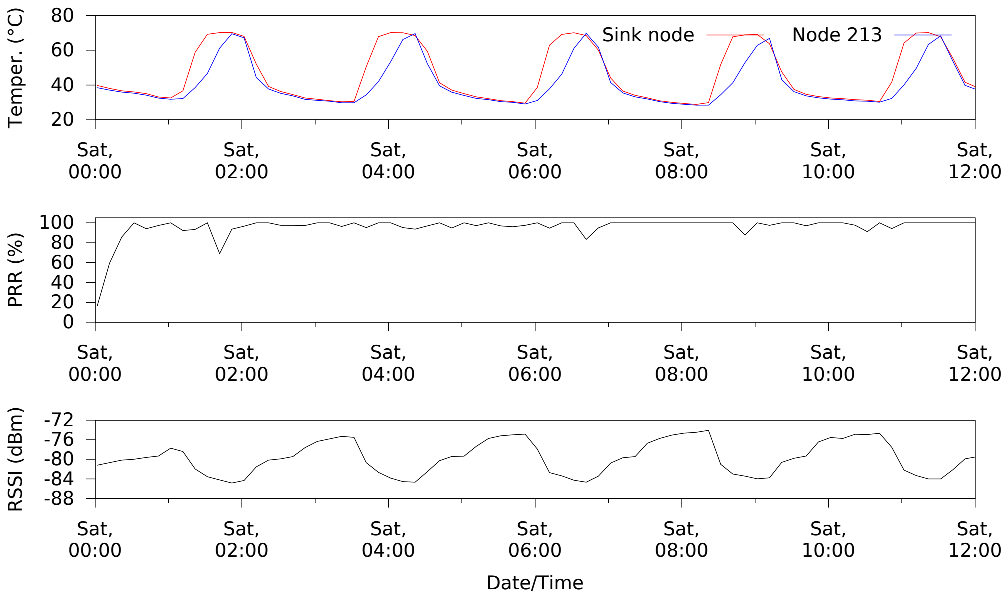
3.4.1 Performance in Augmented Testbed

Using the static protocol parametrization tool developed in WP3, we derive $CCA_{Thr} = -76$ as optimal value when using TempMAC in our TempLab testbed. This value is obtained by feeding the parametrization tool with a packet reception rate of 95% as performance requirement and the temperature profiles shown in Table 3.3. This implies that the sensor network running TempMAC should be able to sustain such performance despite being subject to temperature variations up to 55 °C.

We run our experiments replaying the temperature profiles recorded in Madrid using TempLab's closed loop application (see the folder `D-4.4/temperature/templab/closed_loop` in the attached zip file). To minimize the experimentation time, we time-lapse the trace by a factor of 10, i.e., we replay using TempLab in 12 hours the temperature profiles that actually occurred in Madrid during 5 days (temperature profiles used in TempLab can be found in the folder `D-4.4/temperature/templab/temperature_profiles` in the attached zip file). We then run the data collection application described in the previous sections using TempMAC with $CCA_{Thr} = -76$ on all the wireless sensor nodes that are equipped with infra-red heating lamps.



(a) Link between node 200 and 203



(b) Link between node 200 and 213

Figure 3.12: Performance of TempMAC in our TempLab testbed ($CCA_{Thr} = -76$ dBm). Using RELYonIT solutions, we can meet the specified performance goals (PRR $\geq$ 95%).

Table 3.4 shows the results. The network delivers successfully more than 97.4% of the packets when using TempMAC with $CCA_{Thr} = -76$, hence showing that the newly-designed protocol meets the desired performance goals. The reception ratio is actually higher than 95% on each of

Node ID	Average PRR	Minimum PRR
202	97.24%	60.00%
203	97.74%	70.83%
205	98.30%	81.82%
208	97.87%	62.50%
209	96.73%	45.45%
213	97.19%	77.78%
Network	97.51%	–

Table 3.5: Performance of ContikiMAC with an initial CCA threshold selected using the parametrization tool developed within WP3 ($CCA_{Thr} = -85$ dBm). The experiment in our TempLab testbed when replaying the same temperature profiles recorded in Madrid shows that when using RELYonIT solutions, we can meet the specified performance goals (PRR $\geq$ 95%).

Node ID	Average PRR	Minimum PRR
202	97.60%	58.33%
203	51.16%	6.67%
205	14.98%	5.56%
208	98.78%	64.29%
209	95.69%	8.33%
213	39.99%	5.88%
Network	66.20%	–

Table 3.6: Performance of default's ContikiMAC ($CCA_{Thr} = -76$ dBm) in our TempLab testbed when replaying the same temperature profiles recorded in Madrid. Without using RELYonIT solutions, we cannot meet the specified performance goals.

the link in the testbed, as predicted by our parametrization tool. Figure 3.12 shows a close-up of the performance on individual links: we are able to sustain a high packet reception regardless of temperature variations, in contrast to the results obtained without RELYonIT improvements.

We then run an experiment in which we use the parametrization tool to compute the optimal CCA_{Thr} for ContikiMAC. We specify a packet reception ratio of 95% as in the previous case and obtain $CCA_{Thr} = -85$ dBm, i.e., our tool guarantees us that such threshold is low enough to avoid any intersection with the signal strength of the received packets with the specified temperature bounds [4] and hence that the network can sustain the desired performance also at high temperatures. Table 3.5 shows that this was indeed the case. The average packet reception rate in the network using ContikiMAC with $CCA_{Thr} = -85$ dBm was 97.51 %, fulfilling the desired performance requirements. In particular, we have observed that for none of the links the received signal strength has reached values below -85 dBm, showing that the parametrization tool did a commendable job in predicting the impact of temperature variations on low-power wireless communications.

Node ID	Average PRR	Minimum PRR
102	97.33%	89.29%
105	97.85%	92.98%
106	49.86%	1.32%
107	98.00%	91.30%
Network	85.77%	–

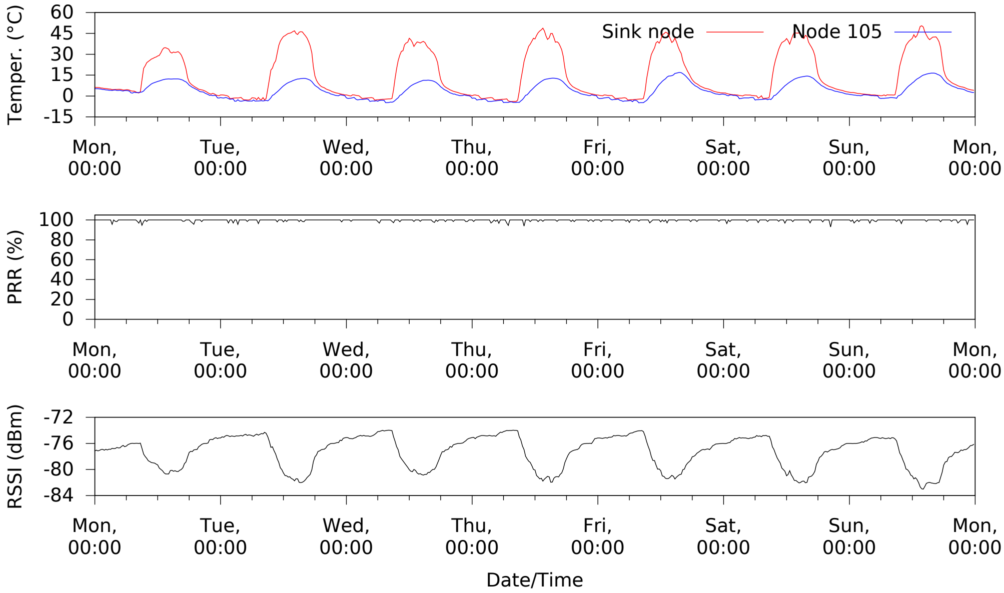
Table 3.7: Performance of TempMAC at the DEMOPARK facility in Madrid (initial threshold $CCA_{Thr} = -77$ dBm). The experiment took place at the end of December 2014 and shows that we can meet the specified performance goals.

We further run ContikiMAC using $CCA_{Thr} = -76$ dBm using the same settings as above, and observe a packet reception rate in the network of only 66.20%. This is in line with the experiments shown in Figure 3.10 (Table 3.6 contains a detailed summary of the packet reception on each of the link in such scenario), and three out of six links (i.e., the links between the sink and nodes 203, 205, 213) exhibit a low delivery rate. This confirms that only when using the RELYonIT toolchain the performance requirements needed by the ACCIONA industry partner can be successfully met.

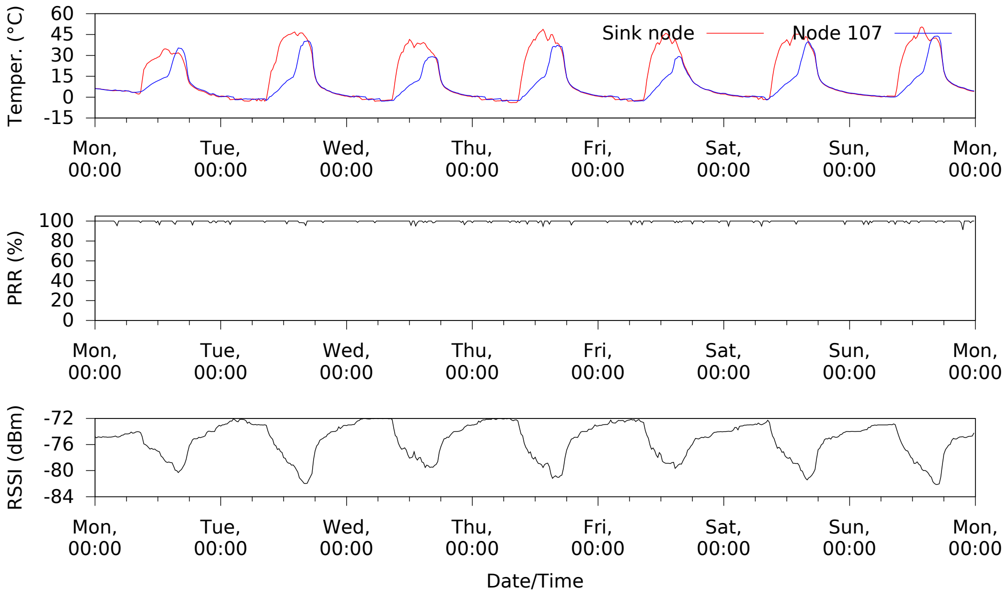
3.4.2 Performance at DEMOPARK in Madrid

We carry out the same experiment at the DEMOPARK facility in Madrid. Using the static protocol parametrization tool developed in WP3, we first derive the optimal value of CCA_{Thr} that should be used by TempMAC to obtain the required performance. The parametrization tool performs a prediction of the RSSI attenuation in each of the links in the deployment based on the connectivity information provided as input. In line with the previous experiments, the tool warns us that the signal strength of nodes 103 and 104 is too weak and these nodes are unable to reliably communicate even at low temperatures. Coherently with Section 3.2, we therefore do not include these links in our experiments and we filter them out when computing network statistics. The parametrization tool further informs us that the network can achieve an average PRR of 85% when using TempMAC with $CCA_{Thr} = -77$, but that the link between node 101 and 106 cannot sustain a high PRR at high temperatures, as the strength of the signal is too close to the noise floor (and the radio is hence unable of reliably decoding the packets – regardless of the initial CCA threshold chosen).

Table 3.7 shows the PRR in the network during a long-term experiment in the beginning of January 2015. The average PRR in the network is higher than 85%, hence within the specified performance requirements. As expected, all links except the one between node 101 and 106 sustain a reception ratio close to 100% despite variations in ambient temperature. Figure 3.13 shows in detail the performance of two of the links in the network. The difference with the performance of traditional techniques shown in Figure 3.8 are quite evident. TempMAC, indeed, adapts the CCA threshold at runtime as shown in Figure 3.15 and automatically avoids the wake-up problem at high-temperatures highlighted in [4]. In particular, Figure 3.15 shows the different CCA settings over time on the link between node 101 and 102: please note that each



(a) Link between node 101 and 105



(b) Link between node 101 and 107

Figure 3.13: Performance of TempMAC at the DEMOPARK facility in Madrid (initial threshold $CCA_{Thr} = -77$ dBm). The experiment took place at the end of December 2014 and shows that when using RELYonIT solutions, we can meet the specified performance goals.

(a) (b) (c) (d)

Figure 3.14: The presence of other buildings shadowing the sunshine introduces largely different on-board temperature profiles among different wireless sensor nodes.

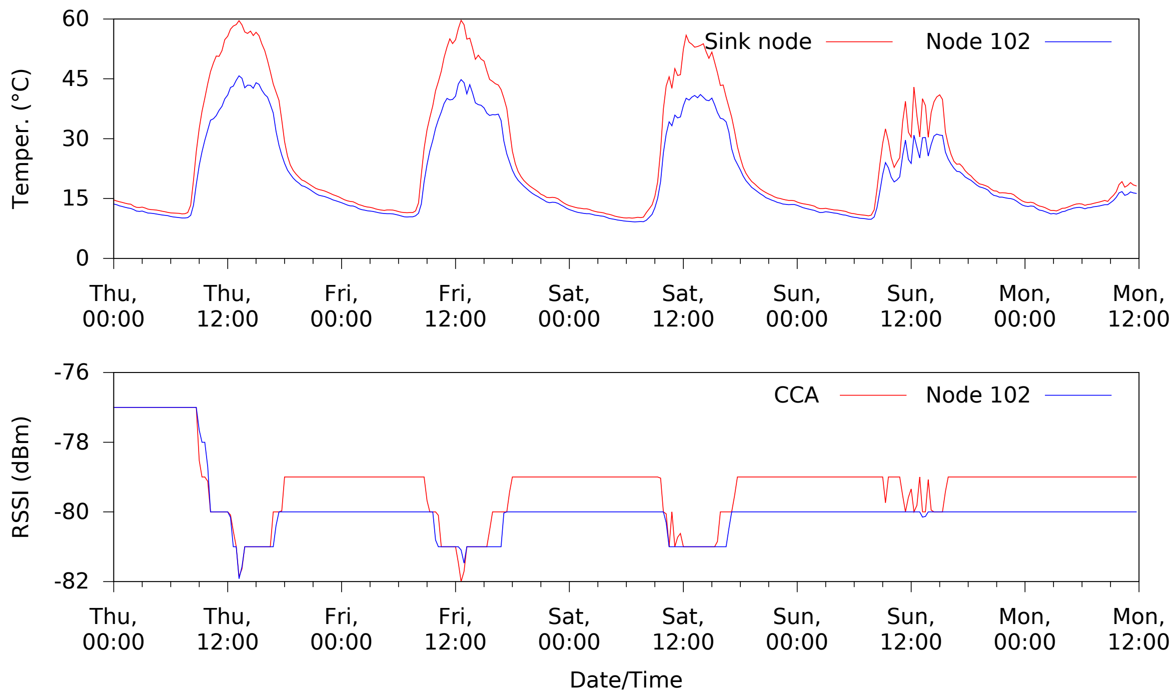


Figure 3.15: TempMAC adapts the CCA threshold based on the temperature measured locally on the sensor node and on the highest temperature recorded in its neighbourhood.

node maintains the highest temperature recorded in the neighbourhood (i.e., there is no ageing of ΔT_{tx}), and therefore the CCA threshold does not return to the initial value of -77 dBm during night-time.

Please notice that in Figures 3.13(a) and 3.13(b) the temperature profiles of node 101, 105, and 107 are rather different. This is because in our deployment scenario, the sun does not heat all sensor nodes equally, but rather only a few of them at a time as an effect of the shadow generated by the surrounding buildings (see Figure 3.14).

We further carry out an experiment employing ContikiMAC using the initial CCA threshold

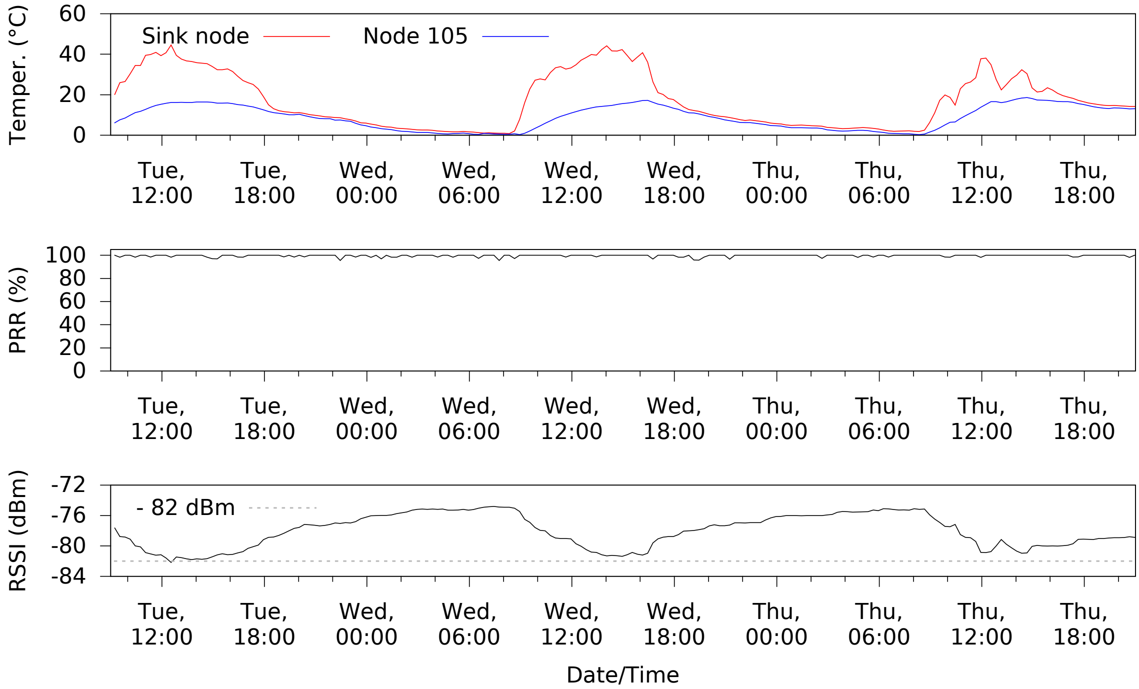


Figure 3.16: Performance of ContikiMAC at the DEMOPARK facility in Madrid when selecting the initial threshold $CCA_{Thr} = -82$ dBm according to the output of the parametrization tool developed in WP3. The experiment took place at the end of January 2015 and shows that we can meet the specified performance goals.

specified by our parametrization tool. The latter guarantees a performance higher than 95% when using ContikiMAC with $CCA_{Thr} = -82$ dBm on nodes 102, 103, 105, and 107, with the link between the sink and node 105 being the closest one to exceed the -82 dBm value at high temperatures. Our results confirm that the network formed by nodes 102, 103, 105, and 107 can indeed sustain a performance of 99% (requirements met). Figure 3.16 further shows that the signal strength of the link between the sink and node 105 never intersects CCA_{Thr}, confirming the hints given by our parametrization tool.

3.4.3 Runtime assurance and adaptation

We finally evaluate the reliability of our runtime assurance component by using controlled settings to produce temperature values outside the specified model bounds. Towards this goal, we isolate the temperature profiles recorded in Madrid during a cloudy day and during a sunny day.

We compute our worst-case model parameters based on the temperatures recorded on a cloudy day (day 1) and use the parametrization tool to compute the optimal initial CCA threshold of ContikiMAC[6] to achieve a packet reception rate of 95%. We then demonstrate

[6]The choice of ContikiMAC as opposed to TempMAC is motivated by the lack of adaptation by the former.

Node ID	Day 1, maximum temp. [°C]	Day 2, maximum temp. [°C]
200	40.90	77.93
202	37.11	63.99
203	37.50	46.83
205	38.68	46.25
208	37.01	71.72
209	38.68	46.25
213	37.01	71.72

Table 3.8: Temperatures used in our example of runtime assurance and adaptation with ContikiMAC.

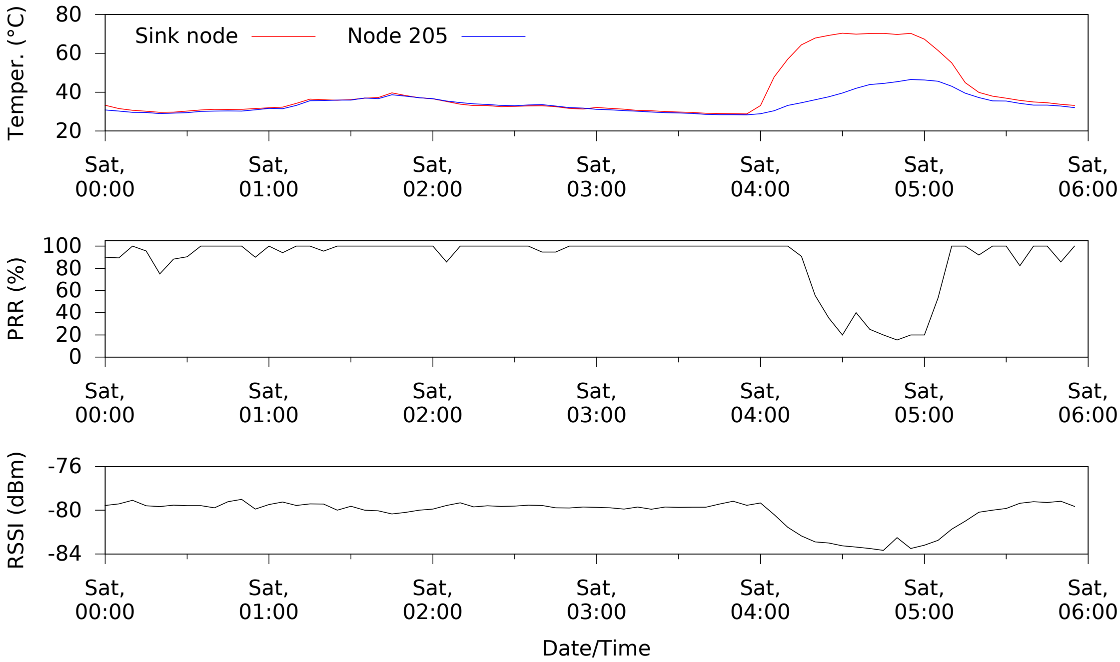
that the network achieves the required performance during day 1, but not during day 2 (sunny day in which the on-board temperature of sensors nodes reached values up to 62 °C).

Table 3.8 shows the temperatures of the two days recorded on each of the nodes. We replay these temperatures using TempLab (the values shown in the table are increased by 17 °C to compensate the lack of cooling enclosures as explained in the previous section) and by time-lapsing the temperature traces with a factor of 8. The WP3 parametrization tool returns that ContikiMAC's CCA_{Thr} should be -81 dBm to guarantee a reliable performance in the presence of the mild temperatures of day 1.

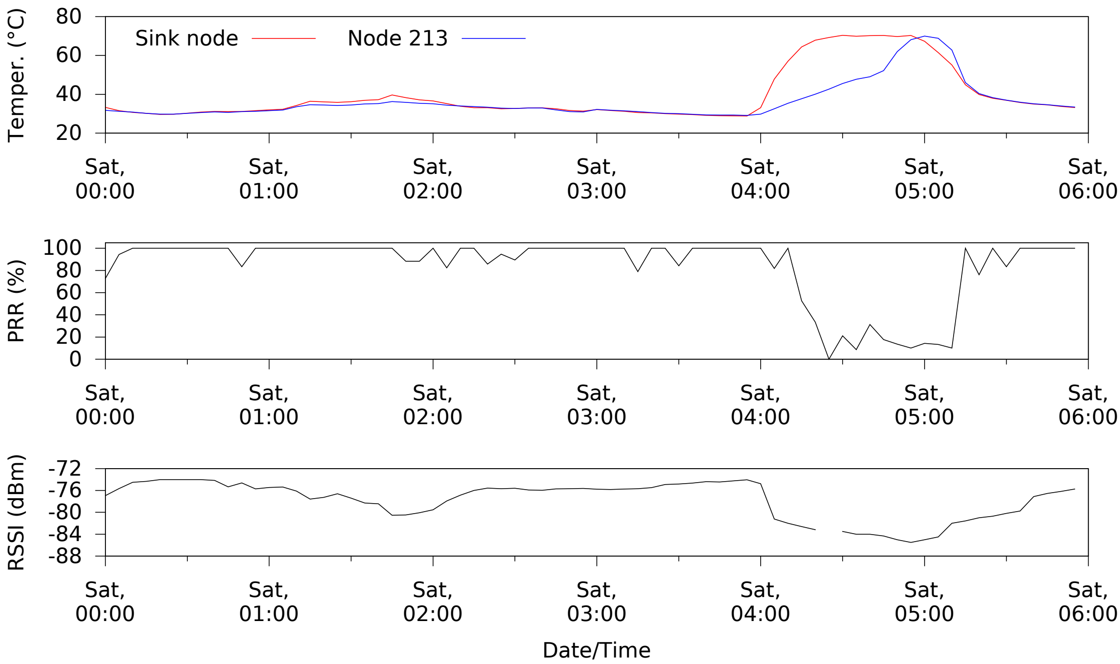
Figure 3.17 shows that, as expected, ContikiMAC sustains the desired performance during the first day, but not during the second day. As soon as the temperature bounds are exceeded, the runtime assurance fires, indicating that the model used to compute the initial threshold is not valid any-more. We therefore re-run the WP3 tool using the temperatures from day 2 and obtain that ContikiMAC should employ a $CCA_{Thr} = -85$ dBm to be able to sustain the desired performance with the temperatures of day 2.

Figure 3.18 shows that, indeed, when running ContikiMAC with $CCA_{Thr} = -85$ dBm, we obtain the desired performance on both days, fulfilling the performance requirements indicated by the ACCIONA industry partner.

TempMAC is an adaptive protocol that adapts the temperature bound at runtime and therefore already embeds runtime assurance.

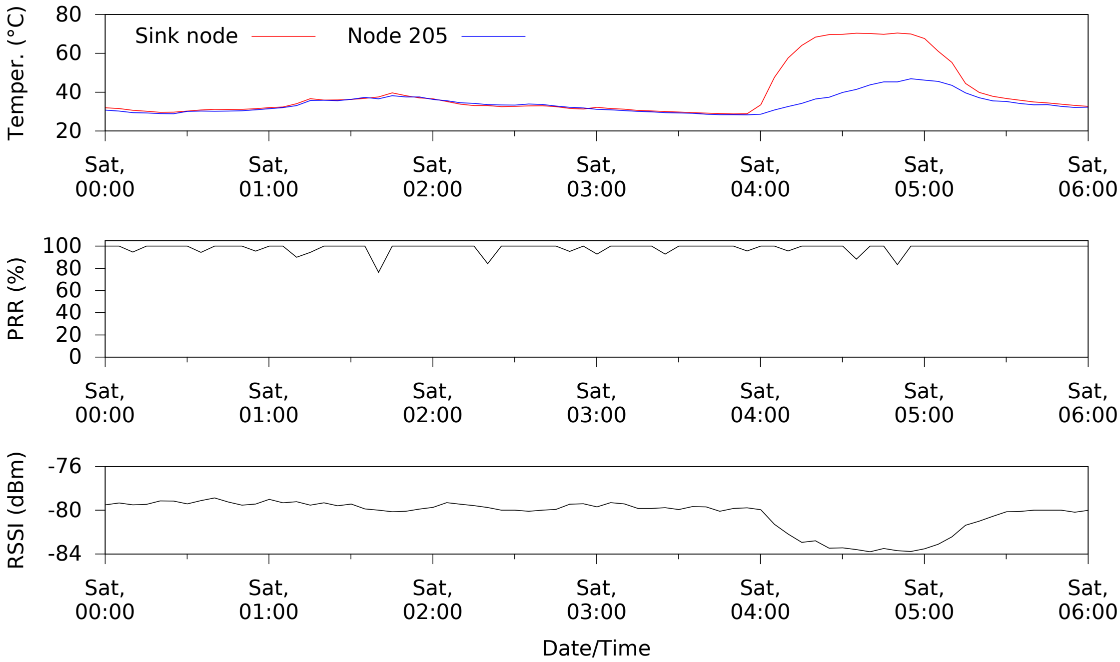


(a) Link between node 200 and 205

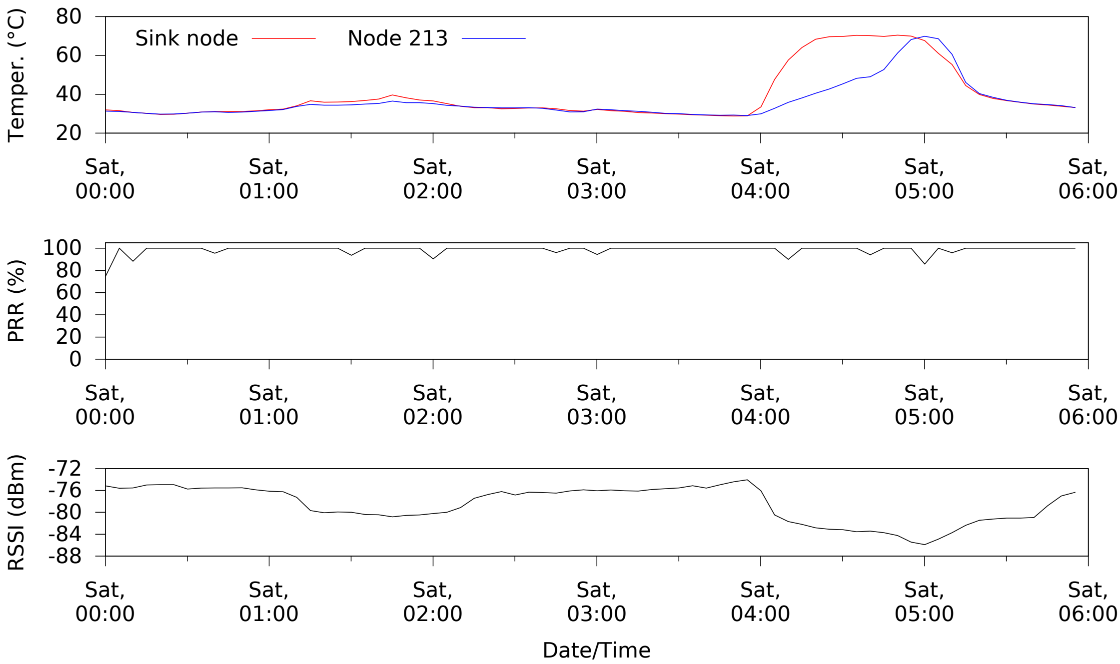


(b) Link between node 200 and 213

Figure 3.17: Performance of specific links in the TempLab testbed when running ContikiMAC using an initial CCA threshold computed on the temperatures recorded on day 1 ($CCA_{Thr} = -81$ dBm).



(a) Link between node 200 and 205



(b) Link between node 200 and 213

Figure 3.18: Performance of specific links in the TempLab testbed when running ContikiMAC using an initial CCA threshold computed on the temperatures recorded on day 2 ($CCA_{Thr} = -85$ dBm).

4 Barcelona Deployment: Interference

This section describes the final integrated experiment in mitigating excessive power consumption due to radio interference. The experiment is conducted in a practical deployment in the Smart City testbed at the 22@Barcelona innovation district, and in a testbed hosted by the University of Lancaster.

4.1 Experimental Setup

Nodes are situated in the tarmac at points surrounding a central base station (shown in Figure 4.1) such that the distance between any node and the sink is the same for all nodes, and each node reports changes in sensor readings back to the central sink. This arrangement ensures that under normal circumstances we would expect to see near perfect packet delivery rates.

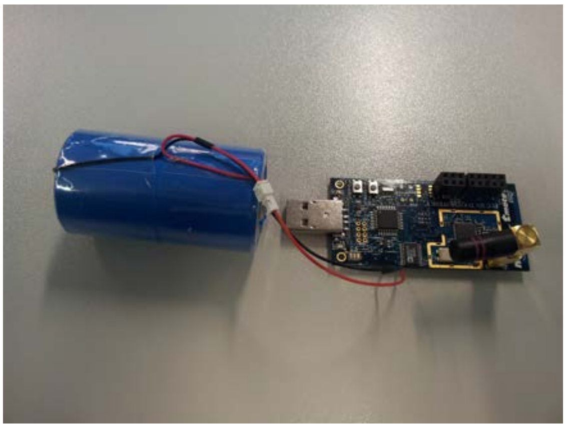
Smart City facility in Barcelona As part of the wider 22@Barcelona innovation district, the deployment consists of 5 sensor spots in a single load/unload zone in the corner of a larger parking area. These spots are used for short stops for loading or unloading vehicles, and as such see a large amount of use with vehicles arriving and leaving far more often than would normally be seen in a traditional parking space.

(a) The sink node, as mounted during the experiments.

(b) The FastPrk area, in normal usage.

Figure 4.1: The experimental area and the sink node in situ.

All nodes except for the sink are embedded in the road surface, and therefore do not receive much interference from other radio sources. The sink, however, is mounted 4 meters in the air attached to a lamp post. It is less than a meter away from a commercial WiFi access point mounted on the same lamp post, as shown in Figure 4.1(a), so it receives significant interference.



(a) TelosB with an attached battery.

(b) Sensor node deployed in the Tarmac, shown with USB a debug cable attached.

Figure 4.2: The sensor hardware deployed in Barcelona, embedded in the road surface.

The nodes used for the deployment are TMotes with a 2.4 GHz Texas Instruments CC2420 radio as shown in Figure 4.2, and are running the Contiki operating system. Batteries normally power each device (see Figure 4.2(a)), but they can also be externally powered for debugging (Figure 4.2(b)) or accessibility reasons.

Packets from the sensor nodes embedded in the road surface are transmitted either when a new event occurs, or periodically to update the sink on the node's status. This entails that the times when data packets are sent are unpredictable, and therefore requires that the sink spends the majority of its time idle listening for incoming data packets. It is this near-constant idle listening state that causes spurious processing to occur when any spurious packets are received from other, 2.4 GHz band devices.

All the nodes (sensor nodes, and the sink node) in the experiment are battery powered, and for the deployment to be a reasonable simulation of a real deployment in a remote location, the lifetime of these nodes must be kept as long as possible. This is limited by the battery capacity, and the number of wake-up events caused by radio packets must be minimised to prevent the sink node from expiring faster than it otherwise should.

Phase 1: Environmental Modelling When the experiment is set up, the environment is sampled from the sink node to determine the baseline environmental noise fingerprint, and the nodes are pre-loaded with this data prior to completing their installation in the road surface and the lamp post. Parameters are then chosen from the available data using the tools created in WP3 to configure the nodes' channel check rate. These are then stored on the nodes.

Once the nodes are configured and running, we run the experiment over a number of hours, and record the power consumption that is caused by spurious wake-up events. The sensor data, along with additional node health information, is also recorded by the sink node, as shown in

Figure 4.3: Parking sensor data graph for a single park sensor; Of note is the high frequency of events, i.e., the high number of vehicles arriving and leaving the parking area.

Figure 4.3 and Figure 4.4.

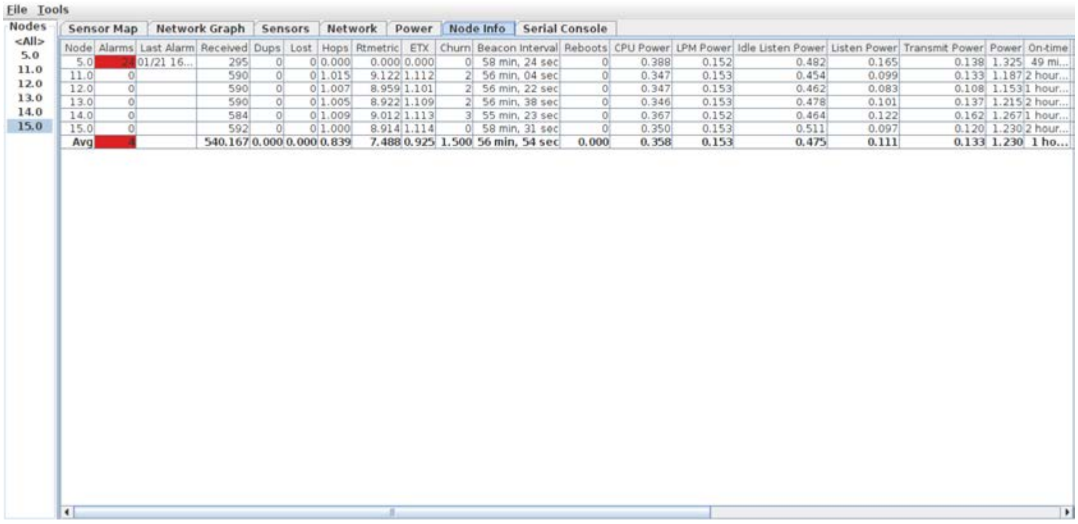
The recorded data is then used to verify that the model corresponds to the environmental factors over a long time duration. For the model to have performed as expected, the energy consumption of the sink node should match what the model predicted it should be at the beginning of the experiment.

Phase 2: Performance Testing Taking this baseline recording of the environment without the use of any remediation measures, we can then intentionally introduce anomalous interference to the system to violate the model. This is done to cause the nodes to detect a model violation, and request that the parameters be changed (namely, the channel check frequency), or the model be updated entirely to match the new environment.

In Figure 4.4 we demonstrate that a node has determined that the parameters it is running with are not adequate enough to mitigate the environmental interference it is experiencing, and it has raised an alarm to signal as such.

When nodes signal a model violation, and the parameters available to alter are insufficient to adjust the node behaviour to match the new environment, our remediation function in ContikiMAC is executed. This re-samples and re-builds the model from environmental readings at the event, allowing the nodes to completely reconfigure their behaviour to match the new interference patterns.

To invoke these events, we inject additional WiFi interference into the environment around the sink, causing the number of wake-up events to increase beyond that which it should be receiving, and thus, breaking the model. This is performed using a standard WiFi card with traffic generated using *iperf*—a network performance measurement tool.



Node	Alarms	Last Alarm	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Idle Listen Power	Listen Power	Transmit Power	Power	On-time
5.0	24	01/21 16...	295	0	0	0.000	0.000	0.000	0	58 min, 24 sec	0	0.388	0.152	0.482	0.165	0.138	1.325	49 mi...
11.0	0		590	0	0	1.015	9.122	1.112	2	56 min, 04 sec	0	0.347	0.153	0.454	0.099	0.133	1.187	2 hour...
12.0	0		590	0	0	1.007	8.959	1.101	2	56 min, 22 sec	0	0.347	0.153	0.462	0.083	0.108	1.153	1 hour...
13.0	0		590	0	0	1.005	8.922	1.109	2	56 min, 38 sec	0	0.346	0.153	0.478	0.101	0.137	1.215	2 hour...
14.0	0		584	0	0	1.009	9.012	1.113	3	55 min, 23 sec	0	0.367	0.152	0.464	0.122	0.162	1.267	1 hour...
15.0	0		592	0	0	1.000	8.914	1.114	0	58 min, 31 sec	0	0.350	0.153	0.511	0.097	0.120	1.230	2 hour...
Avg	**4**		**540.167**	**0.000**	**0.000**	**0.839**	**7.488**	**0.925**	**1.500**	**56 min, 54 sec**	**0.000**	**0.358**	**0.153**	**0.475**	**0.111**	**0.133**	**1.230**	**1 ho...**

Figure 4.4: Node statistics and health view in the integrated GUI.

4.2 Integrated Prototype

The final integrated prototype of the RELYonIT smart parking application encompasses a selection of communication protocols, an environmental model, a runtime assurance module, a runtime adaptation module, and a protocol parameterization tool. As in the first integrated prototype, described in Deliverable 4.3 [18], we have implemented all modules to run within the Contiki operating system. The final integrated prototype is a more complete system, including modifications and additions that are based on the experience gained from the first integrated experiment. Furthermore, we have included protocol parameter optimization and runtime adaptation in the prototype, building on Deliverable 3.2 [15]. The protocols are configured with a set of parameter selections that have been analytically determined based on the protocol models and the application's dependability requirements.

In the following, we will describe the individual components that constitute the final integrated prototype, and highlight the differences to the first one.

4.2.1 Environmental Model Tool

For this final integrated prototype, we use an improved version of the environmental model data collection tool presented in Deliverable 1.2 [6] and Deliverable 4.3 [18]. The tool is executed on a node at the deployment site pre-deployment to collect the necessary environmental data to create instances of the environmental model for the deployment site.

The tool has a recording period and an output period. During the recording, which is approximately one minute in duration, we measure temperature and inference. Temperature is sampled using any available temperature sensor, on the Maxfor MTM-CM5000-MSP motes used for the final prototype this was the external Sensirion SHT11 sensor. For interference, the radio is used to sample RSSI at a high frequency, recording approximately 2 million samples in 1 minute. Each sample is compared to the CCA threshold to determine if the channel is idle or busy. In addition to using this data to calculate the IDLE/BUSY PDF as described in Deliverable 1.2, the data is also used to calculate a simplified interference model that is represented by the radio of busy to idle.

Once the temperature and interference model has been created, it is output during the output phase. The outputted data is used to evaluate a set of unique models for the deployment site, each given a probability determined by the frequency they were observed. This data can be used by RELYonIT WP3 components to determine appropriate protocol and parameter selection.

The source code for the environmental model data collection tool can be found in the `D-4.4/interference/env_collect` directory of the attached zip file.

4.2.2 Network Architecture

The network architecture in this integrated experiment consists of multiple protocols that have been selected and optimized to satisfy the dependability requirements of the smart parking application. At the physical layer and the link layer, we use standard IEEE 802.15.4 communication. Atop the link layer, we run the ContikiMAC [8] radio duty cycling protocol and the Carrier Sense Multiple Access (CSMA) MAC protocol. This setup is equivalent to using MiCMAC on one channel, as described in Deliverable 4.3 [18]. The radios of all nodes are configured with a fixed CCA threshold of -77 dBm.

At the network layer and above, we use the Rime communication stack [9]. Rime is a lightweight communication stack that provides an extensive set of communication primitives upon which one can build sensor networking applications. Unlike in the first integrated prototype, we do not employ temperature-aware routing. The reason for this exclusion is that the first integrated experiment revealed that temperature-aware routing has marginal benefit. Instead, the final integrated prototype contains a regular routing layer, as provided by the Rime stack. Our prototype of the smark parking application is programmed using Rime's data collection primitives.

4.2.3 Application Software

Our smart parking prototype is divided in three main components: the CollectView application, the parking client firmware, and the parking sink firmware. The CollectView application is written in Java and runs on a gateway computer, which receives messages from the sink over a serial port, or possibly through an SSH tunnel. CollectView forwards parking slot changes to Worldsensing's web interface, which can be seen in Figure 4.5. CollectView also shows network-wide statistics, which are extracted from the incoming packets from the clients and from the sink itself.

At the node level, the application consists of a sink and a number of client nodes, both of which are implemented for the Contiki operating system and written in the C programming language. The sink is responsible for receiving the data over radio and forwarding it to the CollectView application, which is an external Java application that presents network-wide statistics in a graphical user interface. The clients' main function is to send information about the status of parking slots. In our experiments, the nodes are not equipped with sensors that determine this status, so we emulate the detection of cars by semi-periodically toggling a variable that represents the parking status upon the expiration of a random timeout. Furthermore,

Figure 4.5: Parking slot status in the Barcelona test location, presented in Worldsensing's *Fast-Prk* web interface.

the clients send packets on average every 5 minutes in order to provide updated node statistics regarding energy consumption, environmental conditions, and protocol state. Table 4.1 describes the packet format of the application. The fields are sent in ASCII text format, and the parameters are separated by blank spaces.

4.2.4 Runtime Assurance Component

The runtime assurance component for the final integrated prototype of the RELYonIT smart parking application implements the runtime assurance framework as described in Deliverable 1.3 [7]. The component performs three tasks: violation detection, violation verification, and violation reporting.

Violation detection is implemented as a lightweight function that compares the system's listening idle energy recorded by Contiki's Communication Power module against a specified threshold. If this threshold is breached, the need for verification is signalled and an application callback function is called giving the application an opportunity to perform any necessary processing prior to verification.

For verification, the system is taken offline for one minute whilst radio interference is sampled. Radio interference is measured in the same way as during environmental collection by taking a series of RSSI measurements at high frequency and comparing these to a set threshold. The busy average is then computed to determine the current environmental interference model. This model is then compared with a target model. If the environmental conditions are outside of the specified bounds, the model has been violated.

When the model has been violated, an alarm is triggered. The alarm is represented as a system flag, and is used to trigger runtime adaptation to adapt the system parameters with the aim of bringing the system idle listening energy consumption back to below threshold. The alarm status is also forwarded to the control system with other system performance data, and

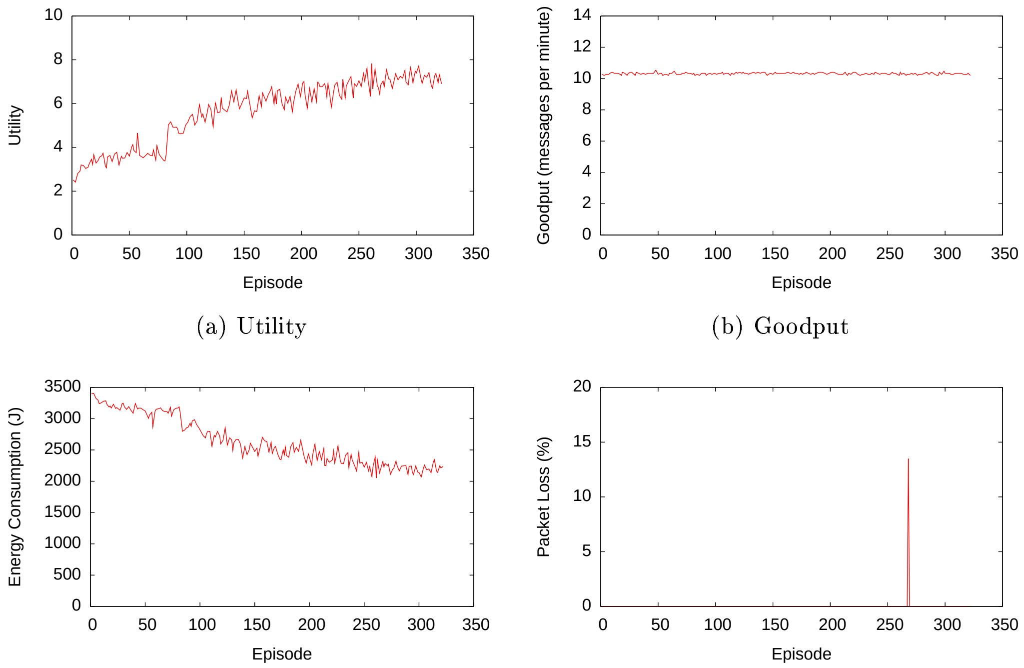
Table 4.1: Packet format for the parking application. Each field contains a 16-bit integer value.

Parameter	Description
Node ID	Node identification number.
SeqNo	Packet sequence number.
Hops	Number of hops from sink.
Clock	Number of clock ticks.
T CPU	Time spent when the CPU is active.
T LPM	Time spent when the CPU is in low power mode.
T Transmit	Time spent in radio transmit mode.
T Listen	Time spent in radio listen mode.
T IdleListen	Time spent in radio listen mode while idle.
Best Neighbor	Node ID of the best neighbor.
Best Neighbor ETX	ETX link metric of the best neighbor.
RT Metric	Routing protocol metric of the node.
Num Neighbors	The number of 1-hop neighbors for the node.
Beacon Interval	Routing beacon interval.
Battery Voltage	Battery voltage level.
Battery Indicator	Battery indicator value.
Light1	Light sensor value 1.
Light2	Light sensor value 2.
Temp	Temperature.
Humidity	Humidity.
RSSI	Received signal strength indicator.
Parking Status	Determines whether a vehicle is parked.
RA Alarm	Runtime assurance alarm.
RA Env Model	Runtime assurance env model.
Adaptation State	Indicates whether runtime adaptation is active.

displayed within Contiki's CollectView application.

Runtime assurance is configured to execute the detection function every minute. This was found to be a suitable periodicity when targeting energy, giving sufficient samples to average out any minor spikes in interference. During an alarm state, however, the detection function cannot be used because idle listening energy consumption can no longer be used to infer changes in interference. The reason for this is that the runtime adaptation is dynamically changing system parameters that affect this value. Instead, during an alarm state, runtime assurance calls the verification function every 30 minutes. This frequency was found to give a balance between the cost of the verification function and the system reaction time to return to a normal state.

A more complete description of runtime assurance can be found in Deliverable 1.3 [7], and the source code for this implementation can be found in `D-4.4/interference/node-software/runtime-assurance.c` in the attached zip file.



(a) Utility (b) Goodput (c) Energy (d) Packet Loss

Figure 4.6: The runtime adaptation component uses a configuration policy learned using simulations of the application. The figures show how progress is made in the learning process according to a utility function, which has been designed with the application dependability requirements in mind.

4.2.5 Runtime Adaptation Component

We provided initial results for the runtime adaptation in Deliverable 3.2 [15]. In this section, we present an extended learning phase that includes more episodes, and a newly designed utility function that focuses on reliability primarily, and energy secondarily. The reason for this prioritization is that it is more important that the parking status is transmitted reliably than preserving the battery slightly longer.

Figure 4.6 shows how the utility function increases as the learning phase progresses. The learning phase is able to improve the configuration policy, as shown by the decreasing energy consumption, while preserving a high goodput and low packet loss. The spike that can momentarily be observed in the packet loss graph occurs when the learning algorithm tries to use a particular configuration that leads to unfortunate consequences. It quickly switches from this configuration policy, however, after it observes the poor results.

In the end of the experiment, we have reached a configuration policy that can be used in the runtime adaptation component on the nodes. Since the learning algorithm has covered many harsh environmental conditions outside the environmental model, it is more likely to satisfy the application's dependability requirements than the configuration selected by the protocol parameterization tool once the alarm state is entered. In Section 4.3.2, we further evaluate runtime adaptation when it executes in the nodes in ULANC's testbed.

4.2.6 Parameter Selection

The demonstration employs the same static protocol parameterization tool as used for the first demonstration scenario described in Section 3.3. A detailed description of the software and the intended work flow can be found in Deliverable 3.2 [15]. This model employs the Radio Energy Prediction model developed in WP2, which is also shipped with the parameterization framework, and can be found in the folder `D-4.4/interference/parametrization_tool/models/energy`. This model also integrates a suitable radio environment model developed in WP1 and a configurable platform model that already fits the nodes employed in the demonstration.

In contrast to the earlier introduced TempMAC model, this model does not directly use trace data collected in the application. Instead the collected traces are fed to a preprocessor that generates a mapping between selectable check rates and the corresponding duty cycle based on the encountered environmental conditions. To enable probabilistic constraints, the tool takes traces from time frames with different characteristics into account and generates individual mappings for each of these distinct time frames. Each time frame is also associated with a relative probability of its applicability. The actual energy model employs the mapping to derive a prediction of the energy consumption based on a specific set of parameter settings. It is employed within the optimization process to find an optimal check rate setting that is capable of meeting the specified goals or constraints on energy consumption under the given environmental conditions.

4.3 Results

4.3.1 TWIST FIRE Testbed: Effects of Interference on Idle Energy Consumption

We follow a multi-phase approach in our evaluation. For the first phase we require a controlled environment (i) to investigate the effects of channel interference on idle energy consumption and (ii) to evaluate the idle energy consumption model presented in Deliverable 2.2 [24]. Towards this end, we use the TWIST testbed, which is part of the FIRE initiative.

The TWIST testbed has $\approx$ 100 nodes, and the set up of the experiment was defined as follows. First, we identified the node that was the closest to the center of the network. This *central* node plays the interferer role and runs JamLab [1] to generate the required interference patterns. The other nodes measure their idle listening time under the various interference patterns generated by the central node. The reasoning behind choosing the central node as the interferer was to maximize the amount of information at the receivers to validate our models.

It is important to notice that to validate our models we only require the overall percentage of channel interference rather than a detailed PDF describing the interference pattern. We generated 18 interference levels, from 5% to 95% in steps of 5%. The interference patterns were obtained by duty-cycling the radio of the node running JamLab. Each interference level was run for 5 minutes. The channel check rate for the ContikiMAC protocol running on the rest of the nodes was set to 8 Hz.

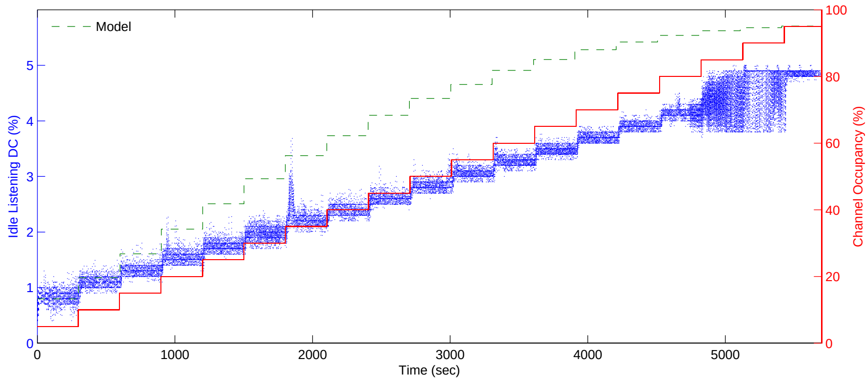


Figure 4.7: Measured idle listening time versus on-time for different levels of channel occupancy.

In Figure 4.7, we plot the interference level generated by the jammer (red line), the idle listening time of the various receivers (blue lines), and the expected result from the model (green line). Idle listening time follows a linear trend with interference, increasing with each increase in interference. With a channel check rate of 8, the minimum on-time is 0.471% and maximum 5.709%, which these results fall between. The model fits the results well at lower interference and starts to loose accuracy at higher interference levels.

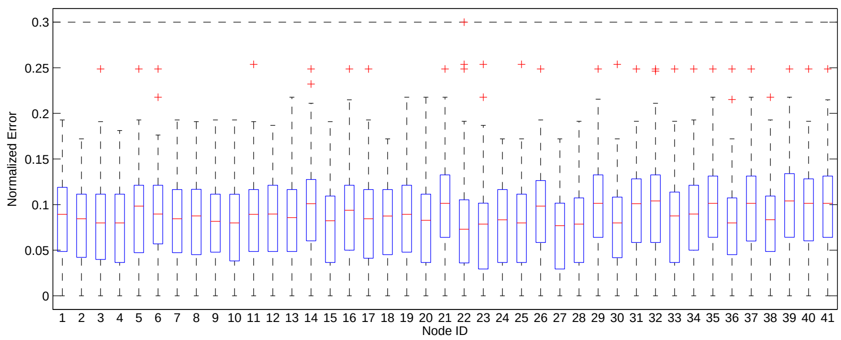


Figure 4.8: Distribution of the normalized error per node.

In Figure 4.8, we show the distributions of the normalized error for each receiver node. The normalized error was calculated as $(\frac{s-m}{m})^2$, where s is the sampled idle listening time, and m is the idle listening time expected by the model. Overall, the mean error across all nodes was 8.36%.

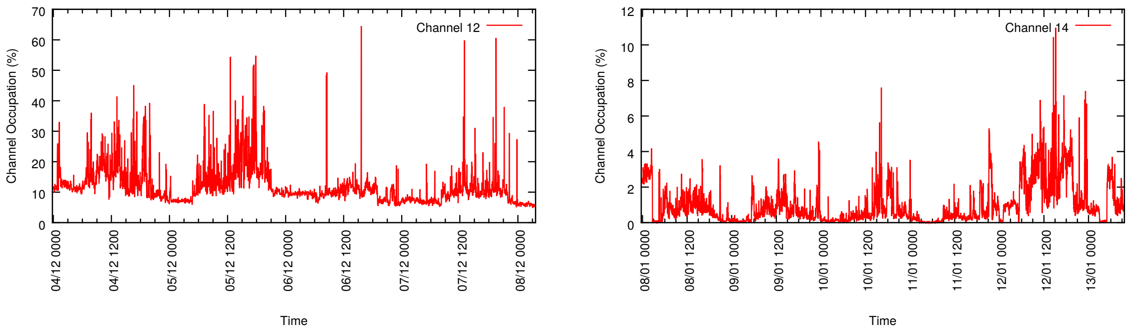
The model has been shown to predict idle listening time accurately at interference levels below 25%. This covers the majority of deployment locations. When the interference is above 25%, the impact is catastrophic to the communication of low-power IEEE 802.15.4 radio technologies.

4.3.2 Local Testbed: Component Evaluation

We utilised both of our testbeds at Barcelona and Lancaster to evaluate the different components of the RELYonIT system specific to the parking application. The components examined separately include the Environmental Model, Protocol and Parameters Selection, Runtime Assurance, and Runtime Adaptation.

Environmental Model

We evaluated the environmental collection tool on-site in Barcelona. As an input to the idle energy prediction model, *the overall percentage that the channel is occupied* is needed rather than the more complex idle and busy PDFs. The tool was ran over 4.5 days on channel 12 and 5.5 days on channel 14.



(a) Channel 12

(b) Channel 14

Figure 4.9: Traces of interference above 77 dBm captured at the Barcelona deployment on channels 12 and 14.

Figure 4.9(a) illustrates the captured interference on channel 12. We found interference to average 14%, with significant bursts of interference occurring during the working day. The weekends were found to have fewer bursts of interference. Figure 4.9(b) illustrates the captured interference for channel 14—this channel was found to have significantly less interference than channel 12. Channel 14 has an average interference of less than 2%. During 98% of the time the interference was less than 5%, and during 99.9% of the time the interference was less than 10%. A PDF of each of channel's interference can be found in Figure 4.10, illustrating the shape of the captured interference.

As this channel was more stable than channel 12, we selected it as the default channel for the evaluation of the integrated prototype.

Protocol and Parameters Selection

For this evaluation, the protocol was statically selected as ContikiMAC, and we used the idle energy model presented in Deliverable 2.2 [24]. The tool requires the environmental model, the target duty cycle, and the energy model to operate. To derive a target lifetime, we examined

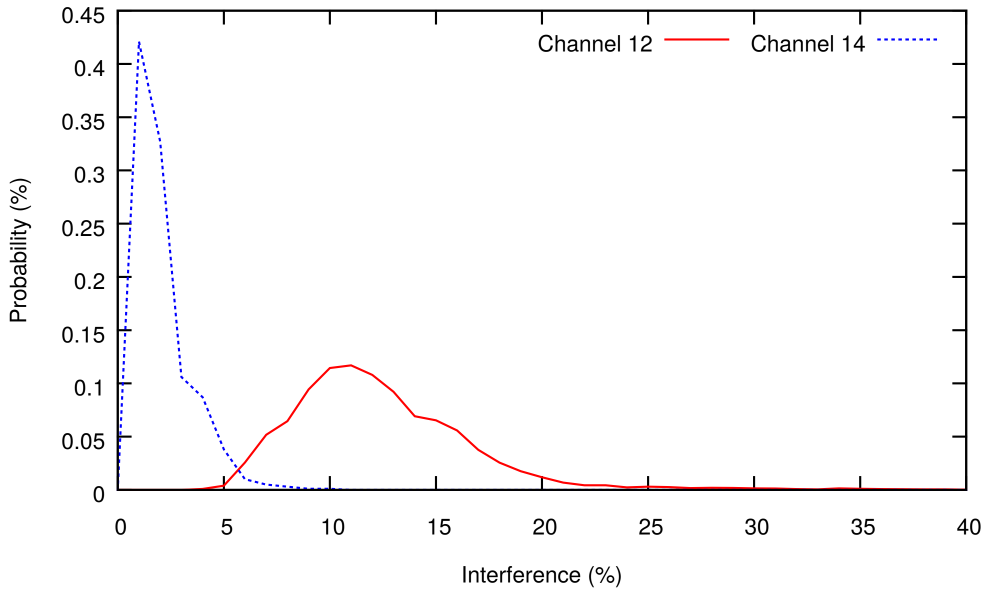


Figure 4.10: Probability Distribution Function of the sampled interference on channels 12 and 14.

the use case requirements. The target lifetime for a Barcelona deployment, was 4 months for the *must* requirement, 8 months for the *should* requirement, and 18 months for the *could* requirement. We decided to focus on the *could* requirement of 18 months.

The use case also specified the battery capacity of 7200 mAh. We assume this cqapacity will be used by both the sensor and mote with a 50:50 split between the cost of sensing and the cost of communication. This gives a capacity of 3600 mAh, with a target life time of 18 months. Whilst communication the energy consumption of the device is approximately 20 mA. As the application generates low-rate traffic, with messages arriving with an inter arrival time of on average greater than 2 minutes, the actual cost of transmitting and receiving message can be ignored as the dominant cost will be idle listening. Using these figures we can calculate the target on-time of the device. This was evaluated to 1.38% for a battery capacity of 3600 mAh a target life time of 18 months and communication energy cost of 20 mA.

The input provided to the protocol and parameter selection tool consisted of the environmental model captured previously for channel 14, and the target on-time of 1.38%. The tool could then use this information to select the channel check rate, which determines how often ContikiMAC checks for communication. The available values are 2, 4, 8, 16, and 32. The tool selected a channel check rate of 16.

An experiment to evaluate how effective the selected value is was performed at the Barcelona facility with only the sink node and no clients. The experiment was conducted over 7 hours during the evening, when interference was found to be more stable and less bursty. Figure 4.11 illustrates the idle listening on-time over this period. As can be seen from this figure, the idle listening on-time is 1.23% on average, which is close to the target of 1.38%. On four occasions, this target is breached during what should be stable interference during the night. As the

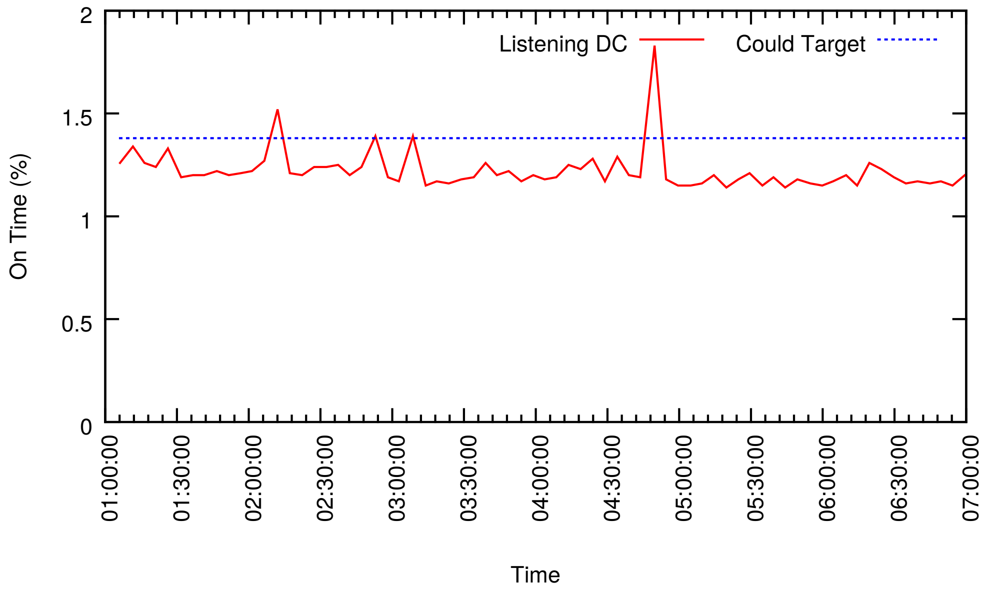


Figure 4.11: Idle-listening duty cycle with a channel check rate 16 at the Barcelona deployment.

message rate of the application is low, and the latency requirements are also low, we decided to investigate the next channel check rate of 8.

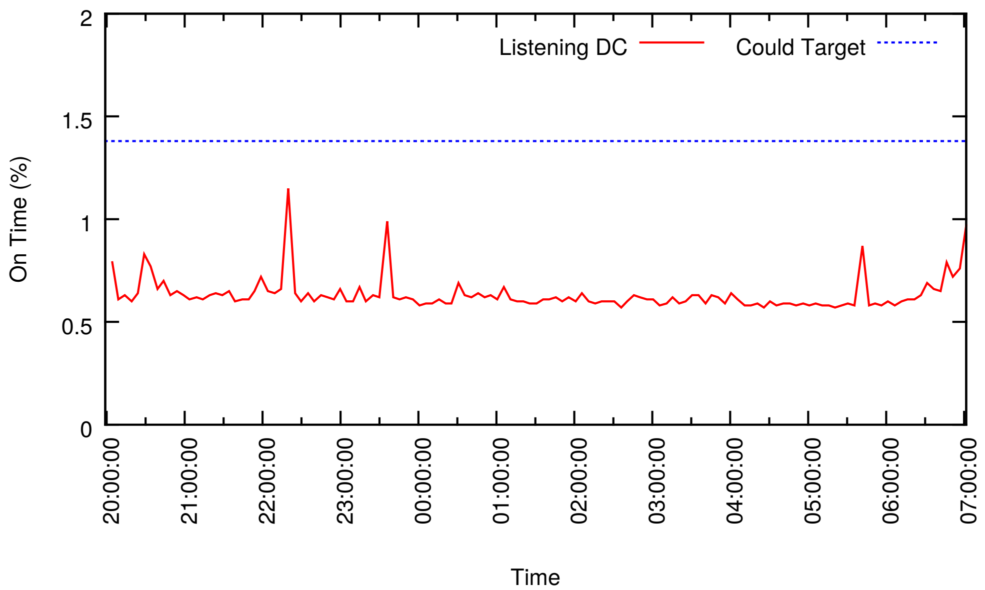


Figure 4.12: Idle-listening duty cycle with a channel check rate 8 at the Barcelona deployment.

Figure 4.12 illustrates the idle listening on-time for a channel check rate of 8. The average idle listening on-time for this setting was found to be 0.64%, which we expect as this is half of the value observed when the channel check rate was doubled at 16. Over a longer experiment of 11 hours, there were no breaches of the target on-time. For the remainder of this evaluation, we have decided to focus on the channel check rate of 8 because this provides more of a safety zone to the target interference line, which will reduce breaches whilst still providing the mechanisms to enable low latency.

Runtime Assurance

To ensure that the interference model is correct for the environment that the nodes are running in, we implemented a runtime assurance design that attempts to detect breaches from the expected environmental behaviour. During the experiment, nodes sample the radio energy and compare the data against a pre-set threshold. If the reading is above the threshold, the node performs a more in-depth sampling, eventually raising an alarm if the radio is experiencing interference.

Nodes check the radio energy at 5 minute intervals, comparing the recorded value against a threshold. If the value is above the threshold, the node proceeds to examine the channel for interference. The interference checking phase runs for one minute, where the node attempts to gather information on the channel. If the node detects an abnormal interference pattern, it enters an alarm state.

From the alarm state, the node only re-checks the channel for radio energy every 30 minutes, returning the node back to the normal running state if interference returns to the model-predicted values. Using this scheme, we should be able to detect and indicate model violations within 5 minutes, and be able to return to a normal running state within 30 minutes of the interference ending.

To test this, we introduce deliberate periodically interference using the WiFi access point and *iperf* to generate anomalous packets, and record when the node detects that the model has been violated. Runtime assurance accurately detected all interference bursts with an average delay of 1 minute and 36 seconds.

Data from the experiment is presented in Figure 4.13, where we see that although the alarm states lag behind the interference, the node successfully detects the model violation.

Runtime Adaptation

We evaluated the effectiveness of Runtime Adaptation through an experiment conducted on the Lancaster Testbed. We conducted the experiment over 11 hours, where we generated artificial interferene for the first 5 hours artificial, whereas the remaining 6 hours had no artificial interference. Interference was generated by means of a WiFi card and an access point configured to share the same frequency as that of the motes. The *iperf* tool was used to attempt to generate a constant amount of interference. One must consider, however, that the lab is situated in a working building, which has other sources of interference that may add to that artificially generated. During the interference period, the system will be in the alarm state, causing the runtime adaptation component will run.

Figure 4.14 examines the idle energy consumption during these two periods. Interference begins just after 18:30, which is signified by the spike in idle energy consumption. At this

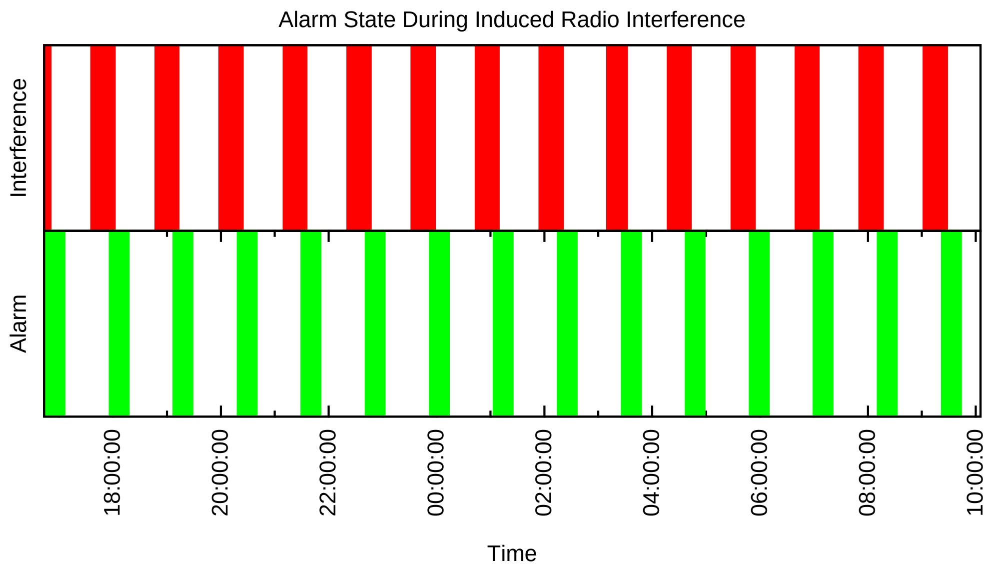


Figure 4.13: Runtime Assurance alarm state during induced interference.

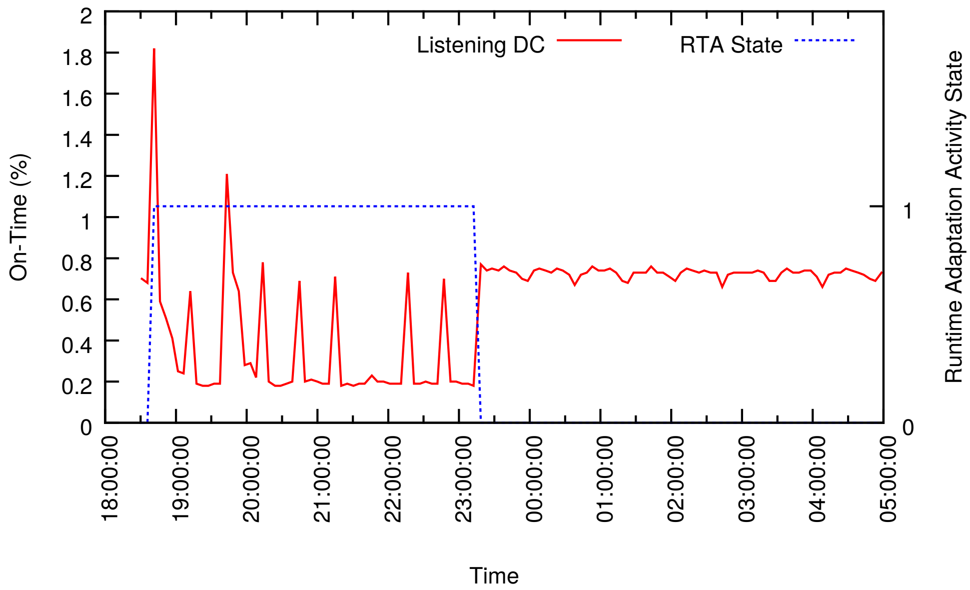


Figure 4.14: Performance of Runtime Adaptation under heavy interference.

point, the model violation is detected by runtime assurance, and an alarm is raised, which triggers runtime adaptation. As we can see whilst runtime adaptation is executing, idle listening drops significantly to the normal behaviour. A large change in on-time is expected due to the available channel check rates, which reduce the on-time by half with each step in selected rate.

During the period when runtime adaptation is active, the on-time averages 0.34% and during the normal period on-time averages 0.56%.

Figure 4.15 illustrates the selected values of the channel check rate during this experiment. During the normal period, the channel check rate is fixed at 8, whilst during the runtime adaptation period the channel check rate is predominantly 2. However, the value of 4 is also attempted sporadically.

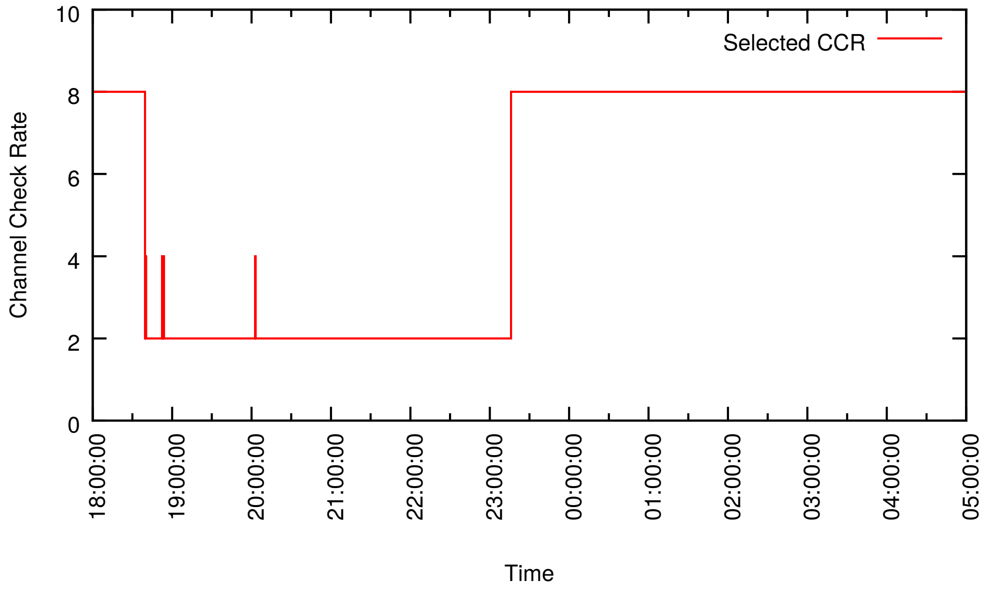


Figure 4.15: Selected channel check rate by the Runtime Adaptation module.

In this experiment, we have shown that runtime adaptation is able to effectively maintain a low radio duty cycle despite being in alarm state. The different spikes in the duty cycle during runtime adaptation shows that the configuration policy causes switches of the ContikiMAC channel check rate occassionally to adapt to new conditions, but that the average duty cycle is maintained at a low level.

4.3.3 Barcelona Testbed: Parking Application Demonstrator Evaluation

In this section, we will present an evaluation of the demonstrator application using results collected during experiments in the Barcelona testbed. Our aim is to achieve the target *could* lifetime of 18 months, which equates to an average on-time target of 1.38%. The demonstrator will include all components presented in Section 4.2, using a channel check rate of 8 and operating on channel 14.

Sink-Only Deployment

The Barcelona testbed facility is a real in-use car park, and as such, programming the nodes in the tarmac can be problematic. The testbed allows for remote sink programming, so to begin our evaluation we examine a network consisting of the sink node only. As idle listening is the dominate energy cost, the cost of receiving parking messages from the clients should have little impact on the results. Examining the sink only will give a cleaner set of results, which may give more valuable insights into how effective the system is. We examined a sink-only deployment with stable interference in Figure 4.12 in Section 4.3.2, which showed that during stable conditions the selected channel check rate would keep the system below the target on-time.

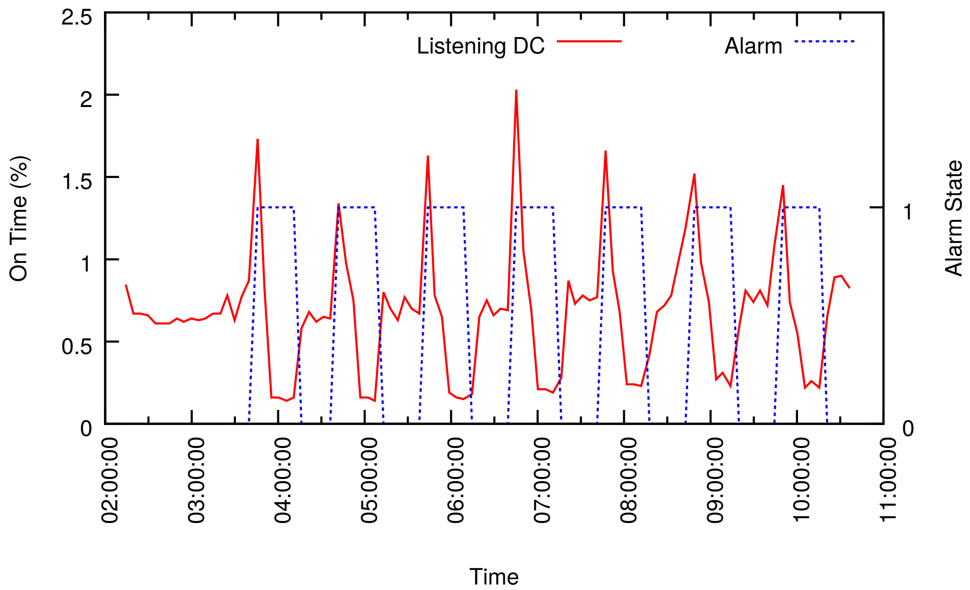


Figure 4.16: Sink-only deployment at Barcelona with a CCR of 8. The sink is affected by periodic artifical interference.

Figure 4.16 examines idle energy consumption under artificially generated interference. Connected to the gateway machine in the Barcelona Testbed is an additional WiFi card, which we use to generate interference similarly to the method used in the Lancaster testbed. We use *iperf* to generate 8 Mbytes of WiFi traffic on the same frequencies as our experiment to simulate an application such as video streaming. Over the course of the experiment shown in the figure, we generate seven bursts of interference, which each last 15 minutes once per hour.

The figure shows that prior to the start of the first interference burst, on-time averaged approximately 0.65%. At the start of each interference burst a spike is seen in idle listening on-time, which is detected by runtime assurance. Runtime assurance is configured to infer that the model is violated when idle energy consumption goes above the target of 1.38%. The model is verified to be violated and an alarm is triggered when interference is above 5%, which was found to be only 2% of the time during pre-deployment.

When the model is violated, runtime assurance will signal the alarm, which will trigger runtime adaptation. Runtime adaptation then dynamically adjust the channel check rate to bring the idle listening on time down. The interference terminates after 15 minutes, but it takes an additional 15 minutes before the alarm is deactivated because runtime assurance will only check interference every 30 minutes during an alarm state to limit the energy cost of checking. During the non-alarm states, the average on-time was 0.67%, and during the alarm states, the average was 0.57%.

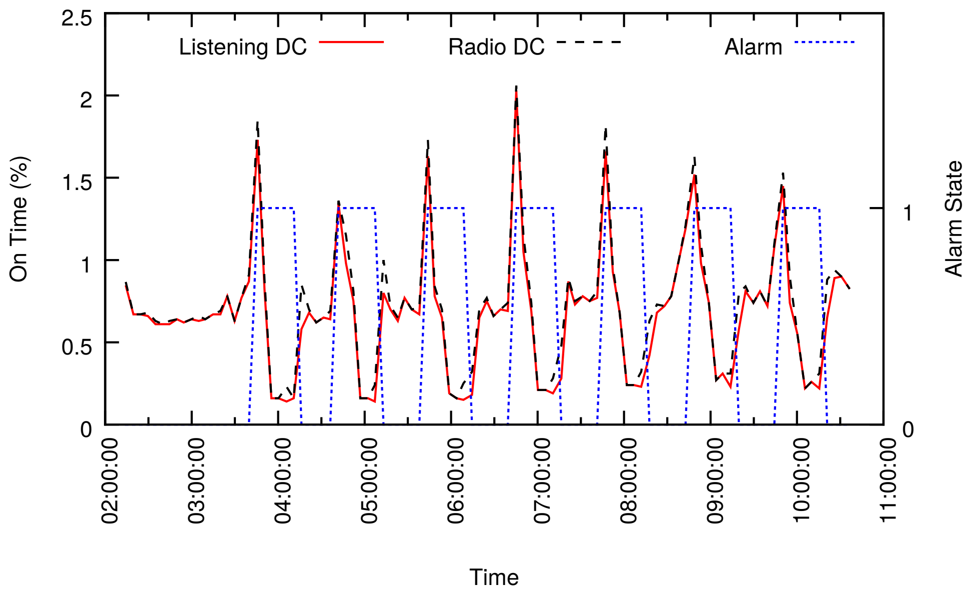


Figure 4.17: Overall radio duty cycle with a CCR of 8, and periodic artifical interference.

In Figure 4.17 we illustrate the complete radio duty cycle alongside the idle listening duty cycle. We see that for the most part, the radio duty cycle equals that of idle listening. These deviate slightly at the start and stop of an alarm because the radio is used to broadcast the alarm flag. In spite of these transmissions, idle listening is the dominating mode that the radio is in when active.

Figure 4.18 illustrates the captured model during the experiment. The captured model is below 5% during when our interference is not being generated. This matches the values captured during pre-deployment collection of the environmental model. During the period when we generate interference, the interference level is approximately 46%, which significantly higher than that observed during the normal environmental conditions.

Full Deployment

In the final set of experiments, we evaluated how effective the RELYonIT system is for the full parking demo deployment. These experiments were conducted in the Barcelona test facility using one sink and five client nodes.

Figure 4.19 illustrates the idle duty cycle of the sink node over the 14 hour deployment. The average idle listening duty cycle for this period was 0.74%. A single runtime assurance alarm was raised. It would appear from the figure that there was no spike in idle listening prior to trigging the alarm. The idle listening values depicted are 5-minute averages (the frequency of CollectView messages). Runtime assurance utilised one minute averages and recorded idle

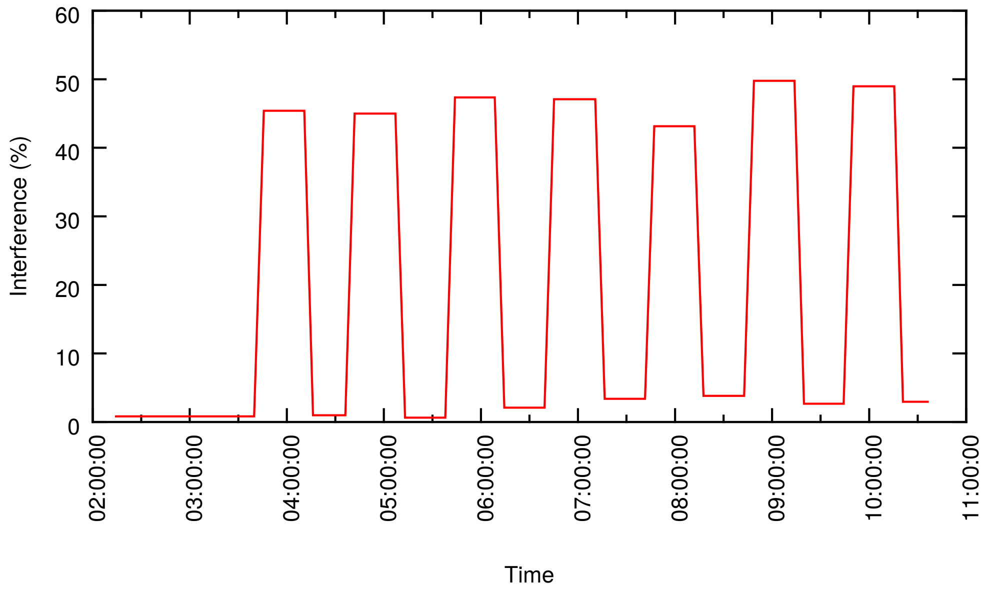


Figure 4.18: Interference model captured by Runtime Assurance during the sink-only test at the Barcelona deployment.

energy as 1.79%, which is above the 1.5% target used by runtime assurance.

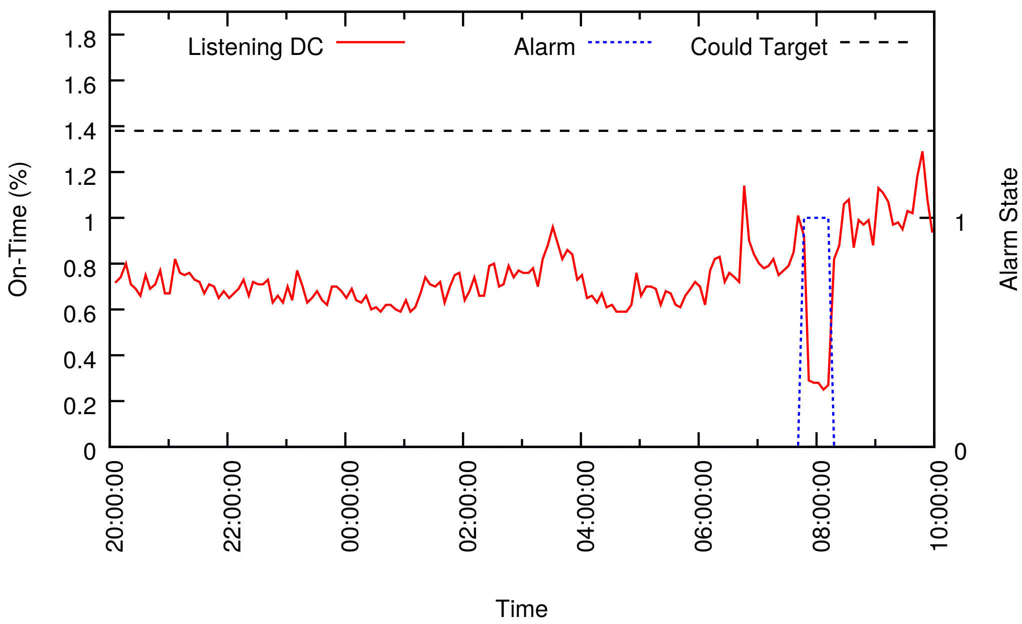


Figure 4.19: Idle-listening duty cycle for the full parking demonstration.

Figure 4.20 illustrates the recorded environmental model during the experiment. The model was only evaluated twice. When the verification was triggered by the increase in idle energy consumption to 1.79%, the interference level was sampled and found to be 7.8%, which is above the runtime assurance target, so a model violation occurred and the alarm was raised. After 30 minutes the model was re-evaluated, and found to be 2.44% below the runtime assurance threshold, so the alarm signal was disabled.

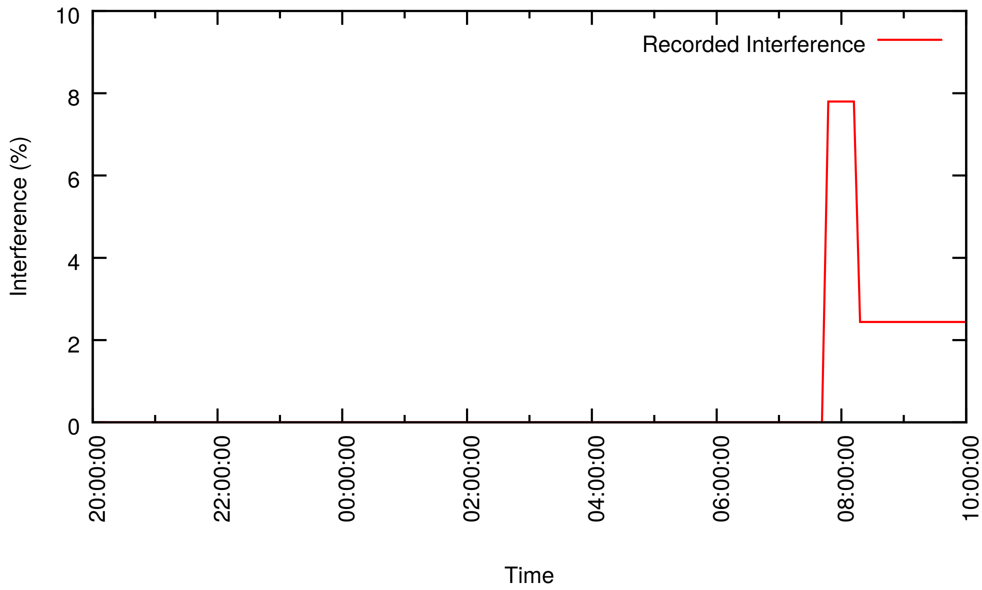


Figure 4.20: Captured environmental model for full parking demostration.

In Figure 4.21, we illustrate the duty cycle of all modes of the radio: idle listening, transmitting, and receiving. Unlike in Figure 4.17, the cost of receiving and transmitting packets can be significant, and our assumption that idle listening is the dominate cost no longer holds. Whilst idle listening averaged 0.74%, receiving was 0.72% and transmitting was 0.35%. These values are higher than expected for receiving and transmitting. We believe that they can be attributed to network connectivity errors and application load, which we discuss later in this section.

Regardless to this higher on-time, the total radio duty cycle averaged 1.42%. This is just slightly over the *could* target of 1.38%. Figure 4.22 illustrates the total radio duty cycle. This shows that quite often the total duty cycle is higher than the *could* target. Whilst this is not ideal, we are more concerned with the average duty cycle, which is close to our *could* target.

Our aim was to have a single-hop network with all nodes directly connected to the sink. All clients but one had a hop count of 1 during the demonstration. Figure 4.23 presents the hop counts for client 37019 and client 247. Client 37019 could not maintain communication with the sink and frequently switched between single-hop and multi-hop communications. As interference for the duration of the experiment except for one period was low, the multi-hop communication must be attributed insufficient transmission range, causing a poor link quality. The client 247 could not be reached for the majority of the experiment, with only a single

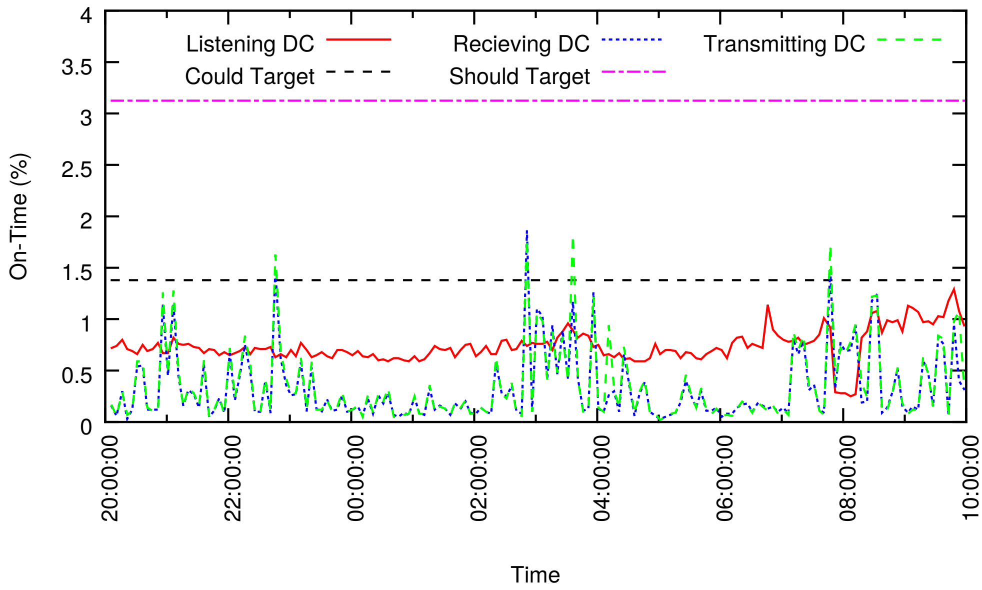


Figure 4.21: Duty cycle of the radio in *listen*, *receive*, and *transmit* modes over the duration of the parking demonstration.

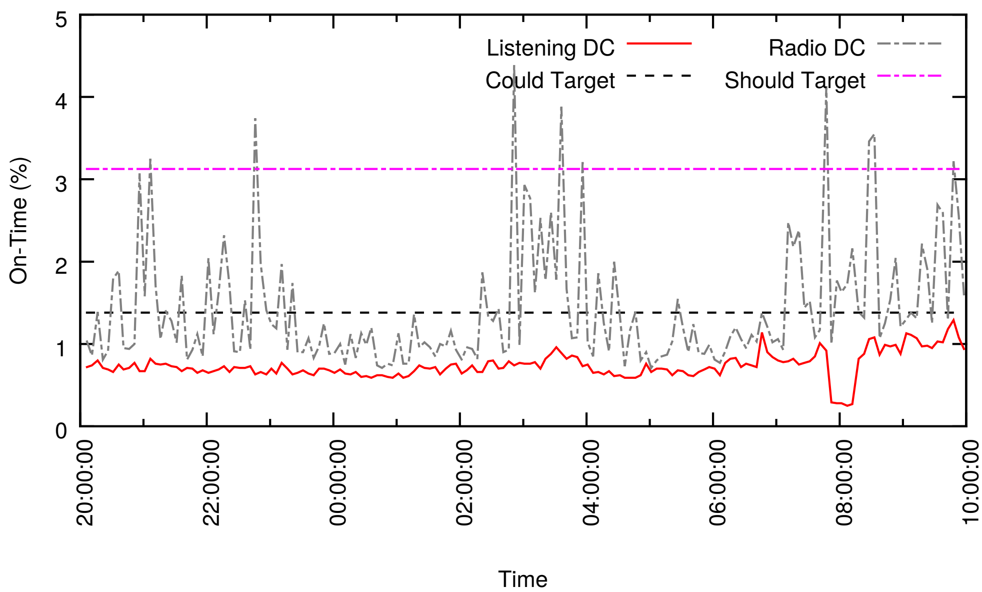
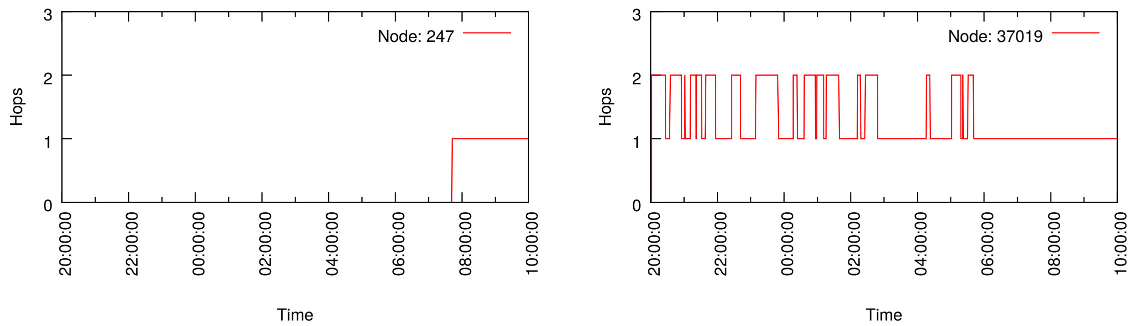


Figure 4.22: Total radio duty cycle over the duration of the parking demonstration.

packet being received around 08:00. Issues with fluctuating hop counts were not seen in local testbed experiments, where the connectivity was better. Logistics made it difficult to better prevision the links of each node.



(a) Client 247

(b) Client 37019

Figure 4.23: Number of hops to the sink for nodes 247 and 37019 during the parking demonstration.

Figure 4.24 illustrates a snippet of the emulated parking events received by the sink over the experiment from a single node, node 3966. We aimed for one parking event every 5 minutes; this is high and would typically not be seen in reality. We felt a higher message frequency would place the system under more strain and act as a better test. Along with parking occupancy messages, statistic messages were sent by each node every 5 minutes to enable our analysis. To enable sufficient reliability, message were sent with default retransmission policy of Contiki's data collection protocol, which is 15 retries.

Even with such a high number of retries configured, we could still observe packet loss. Over the experiment, we had an average packet reception rate of 89% ignoring node 247. Table 4.2 presents in details the loss rate for each node.

Node	**Received**	**Lost**	**Loss (%)**
203	321	2	0.6%
27427	329	23	0%
36966	206	93	11.2%
37019	240	118	38.8%

Table 4.2: Node packet loss.

This poor connectivity, the high message frequency, and the high retransmission policy lead to the additional radio on-time experienced by the sink. Ideally, we would perform another test where the network topology is static, with more realistic traffic patterns and reduced retransmission policy. Unfortunately, this was not possible because of timing and resources. In a deployment without such issues, idle listening would have been the dominate cost as seen with the sink-only experiments. From the results shown, during this experiment the idle listening was found to be 0.74%—well below the *could* target, and this would have enabled a life time of 33.7 months. The total radio on-time was found to be 1.42%, giving an actual life time of 17.6 months, which is slightly below the *could* target of 18 months.

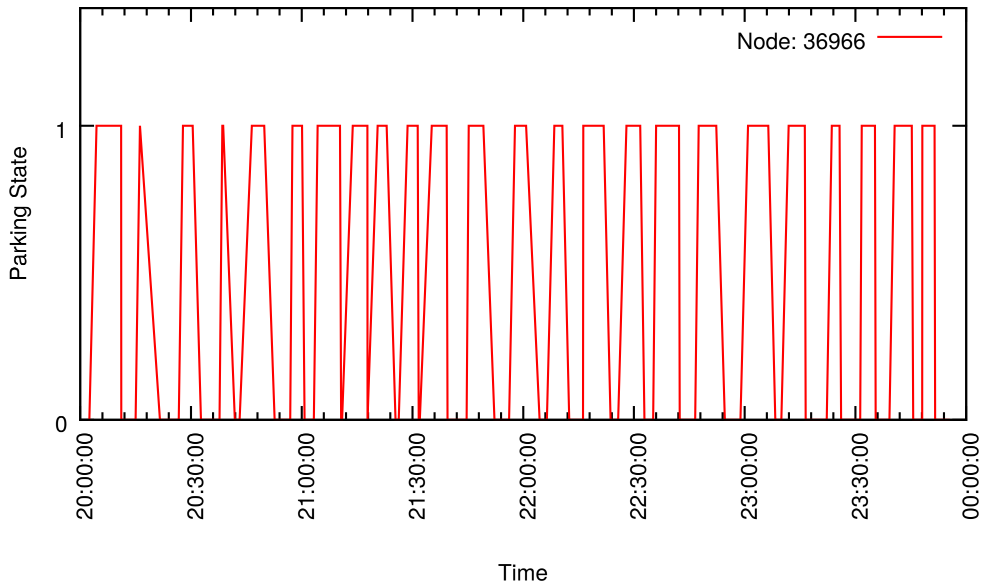


Figure 4.24: Sample of parking events received by the sink from node 3966 during the parking demonstration.

5 Analysis of the Results in Relation to the Selected Use Cases

In this chapter we compare the quantitative results from the experiments described in the previous two chapters against the requirements derived from the use cases.

5.1 Civil Infrastructure Monitoring

To build more energy-efficient buildings, ACCIONA Infraestructures carries out a large number of tests to study the quality of insulating material in existing or newly-built constructions. This is done by means of a wireless sensor network collecting physical parameters such as thermal resistivity, conductivity, inertia, as well as temperature and humidity. Material engineers analyse the tiny changes in the measured variables, and draw conclusions on which materials are more appropriate in different climate areas (e.g., tropical, desert, temperate, or alpine).

Achieving minimal data loss is of utmost importance when carrying out such tests: in case the datasets are non-comprehensive or incomplete, false conclusions may be drawn on whether the insulating materials actually reduce heat transfer. ACCIONA is particularly interested in the ability of the network to sustain a minimal packet loss rate despite high temperature variations. Most networks are indeed deployed on the outdoor façade of buildings, and should be able to cope with the temperature variations typically found in areas with Mediterranean and tropical climate.

One example of such harsh environment is the DEMOPARK facility outside Madrid, a remote site with minimal radio interference, but with daily on-board temperature fluctuations higher than 50 °C. This has been the site of our integrated experiment demonstrating how efficiently RELYonIT solutions can mitigate the adverse effects of temperature on low-power communications.

The experimental results obtained in augmented testbeds and at the DEMOPARK facility in Madrid have shown that reliable communications can indeed be achieved even in the presence of on-board temperature variations up to 55 °C. The experiments have shown that using RELYonIT techniques one can meet not only the 'must' dependability requirement PM-1 (data loss $< 15\%$), but also the 'should' dependability requirement PM-2 (data loss $< 5\%$).

From ACCIONA's perspective, these experimental results have exceeded the expectations, as they show a great improvement in terms of packet reception rate, hence minimizing the number of retransmissions and prolonging the network lifetime. After analysing the results of the project, ACCIONA has decided to incorporate the improvements to the wireless sensor networks that will be deployed in the future material tests at the DEMOPARK facility. As a second step, it has been agreed to run a parallel test at the DEMOPARK facility and on a real building. If the results are positive, the company will adopt the RELYonIT technology in future construction projects.

5.2 Smart Parking

Worldsensing deploys wireless sensor networks in urban environments around the world. In nearly all of these environments, radio interference is present; in many cases interference originates from other networks such as WiFi. RELYonIT has shown that energy consumption of nodes depends strongly on the interference present in the environment. Typical energy consumption patterns of low-power MAC protocols such as ContikiMAC change with interference. Energy consumption in a clean environment can be 10 times less than in a very noisy environment. This means that the estimated node lifetime, and consequently maintenance operations, also might vary by a factor of 10.

For commercial deployments, it is therefore essential that energy consumption can be estimated and that it is possible to control it. RELYonIT has shown that the energy consumption of nodes when affected by a certain degree of interference can be estimated, and that it is possible to control energy consumption so that application targets are met.

An example of an environment with harsh radio interference is at the Barcelona Parking facility, a typical city-centre deployment surrounded by various sources of radio interference. This site was chosen to host our integrated experiment to demonstrate the effectiveness of RELYonIT solutions to control battery life under challenging radio interference.

The results obtained on site at the Barcelona Parking facility, in the TWIST FIRE testbed, and locally in our interference-controllable testbed have shown that the energy consumption of a device can be controlled in spite of high interference to give the desired battery lifetime. The experiments have shown that by using RELYonIT techniques, one can almost meet the *could* dependability requirements SP-13 (lifetime of $>$ 1.5 years) and far exceed the *should* dependability requirement SP-10 (Battery Life $>$ 8 months) and the *must* dependability requirement SP-7 (Battery Life $>$ 4 months).

For Worldsensing it is beneficial to customise communication protocols for the environment in which the sensor network is deployed. The reason for doing so it that it ensures that the expected lifetime of the system can be assessed in specific deployment contexts.

6 Implications of the Results on Future Business Models of the Industry Partners

Based on the results from the final integrated experiments which implement two use cases, the industry partners can make an assessment how the technologies developed in RELYonIT may impact their business models.

6.1 ACCIONA

A study has shown that energy consumption by buildings amount to 40% of the total EU usage, and that buildings generate 1/3 of the GHG in Europe. A lot of effort has been put to improve the current situation, but even new buildings are far from being guaranteed to be energy efficient. Moreover, replacement rate is very low (1% per year), and energy-related renovation rate per year is 1.5%. Historically, the construction industry has had the problem of being slow in adopting innovations. But ACCIONA has been determined to overcome this by applying really innovative solutions in the market. One of them is the use of new materials in buildings. ACCIONA is focusing on the use of new materials/nanotechnology in three fields. One area is advanced insulation systems, which covers the use of aerogels (light, transparent materials). Another area is using nanotechnology to give added value to surfaces, such as ceramics and glass. The third area is focused on improving the ability and performance of perishable construction materials.

The process from creating a new material (or improving an existing one) to using it in the real world is long and challenging. Therefore, adoption of any new technology that can improve the process in terms of cost and time is necessary. As described previously, testing the behaviour of a new material before it is used in a building is a critical issue. A lot of data must be gathered during a time span of weeks to months in order to compare them to previous simulations. And that is the only possible way to know for certain whether the material is suitable or not. Moreover, some tests need to run for several years; in those cases it is mandatory to use a monitoring system capable of being easily deployed in a wide range of locations at the real building. Using wireless sensor networks makes sense to help solving the problem.

The RELYonIT project makes it possible for us to enhance tests of new materials in the harsh environments we are facing. After analysing the results of the project, ACCIONA has decided to incorporate the improvements to the WSN that will be deployed in all future new material testing to be carried out in the DEMOPARK facilities. As a second step, it has been agreed to run a parallel test in our DEMOPARK facilities and a real building. The positive results of this action will lead to the company decide whether to change the procedures in all future construction projects that may involve new materials to adopt this technology.

6.2 Worldsensing

The rise of new radio technologies like SIGFOX, LoRa, Weightless, OnRamp, etc. has changed the landscape of radio solutions for Wireless Sensor Networks markedly: these different new technologies cause a complete disruption on solutions in building up this kind of networks.

In the case of SIGFOX, the technology is closed and proprietary. As a result, users are unable to add new specifications or mechanisms to achieve new demands or requirements [17]. In other technologies, such as LoRa and Weightless, there is an open specification task force or group where associates can introduce new requirements, mechanisms or methods to enhance the current specification [14] [19].

Worldsensing has been studying the suitability of some of these new technologies, with the goal of adding feasible ones to our existing product portfolio. Actually, the company has integrated and is using SIGFOX and LoRa radio technologies in some of its products.

But these new technologies cannot cover the entire requirements of our products. In some cases, the company has detected flaws in each of the radio technologies when trying to enhance our products.

We believe that the results from the RELYonIT project will help us to increase the reliability of these specifications with the inclusion of temperature correction in the first phase. This is because those specifications do not take into account the problems detected and solved in the project related to temperature variations in both sides of wireless communication. An improved product specification comprising higher performance and reliability as well as an extended battery life brings increased value to the product.

On the other hand, the study of the interference patterns in cities (they will become smart cities in the near future) in the 2.4 GHz ISM band will be important for the future development of BitCarrier line of products by the company [21]. These products are designed to detect and record all wireless devices in a certain area, in order to track them to gather information about users habits or traffic patterns inside a city. A current application for these products is to track individual cars in a city and reconstruct the traffic flow, giving information about the ways and paths individual cars are taking. This information helps cities to design new streets or to plan new traffic management policies to smooth vehicle traffic.

The gained expertise in this field will increase the number and type of devices that are detectable. The information gathered by this line of products will allow the company to expand to other markets, such as tracking single individuals in a city by scanning of their portable devices (smart phones, Bluetooth hands free, smart watches, some sort of wearables, etc.). This new family of products can apply to pedestrian pattern detection, branding and advertisements, security, etc., with a huge market space still to be exploited. The company foresees a sales growth in this market of about 100% per year almost for 5 years, with a gross margin about 4.4M€/year for an EBITDA of 3M€ in 2020. With these previsions, the company is planning to hire 4 engineers and 3 sales & operation staff for the Traffic Division.

7 Conclusions

In this deliverable, we have presented our final integrated experiment with the integrated prototype of the RELYonIT system. We have designed the integrated experiment in the context of two use cases provided by our industrial partners. In Madrid, we used ACCIONA's DEMOPARK facility to evaluate the RELYonIT system in a realistic civil infrastructure monitoring scenario. We used Worldsensing's Smart City facility in Barcelona to evaluate our system in an outdoor parking management scenario. Furthermore, we stress-tested our integrated prototype further by using generated interference patterns in the FIRE facility TWIST, and by using generated temperature variations in the RELYonIT testbed TempLab.

For each of these tests, we deployed a slightly different integrated protype: the different scenarios had different environmental models that were learned before the deployment. This difference trickled through the protocol selection, parameterization, and runtime adaptation, as the environmental model affects which protocols and parameter settings are most suitable to select for a particular deployment. The largest difference was in the application itself, where each use case scenario mandates a customized application according to the data delivery requirements.

Our integrated experiment demonstrated the the RELYonIT system provides a communication performance within the bounds of the requirements of the tested use cases. In the interference-heavy Barcelona deployment, we achieved an idle listening duty cycle of 1.42%, which is close to the 1.38% *could* target, leading to a lifetime of approximately 18 months. With further optimization based on the experience of this integrated experiment, it would be simple to reduce the communication cost further, for instance by reducing the frequency of diagnostic messages that are superfluous to the operation of the parking application itself.

At the DEMOPARK facility in Madrid, where high temperature variations were recorded, we were able to correctly predict and mitigate the impact of temperature variations on communication performance. In particular, we sustained a packet reception ratio higher than 95% and we were hence able to meet not only the minimal performance requirements for our application scenario, but also the desired ones.

Through the results of the integrated experiment, we have shown that the integrated prototype of the RELYonIT system can provide probabilistic bounds of communication performance. We have shown that we satisfy the use case requirements from our industrial partners. In this deliverable, we have also described how the results from the integrated experiment—and the RELYonIT project in general—have a considerable impact on the the future business models of the industrial partners.

Bibliography

[1] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga, "JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *Proceedings of the* 10^{th} *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, Apr. 2011, pp. 175–186.

[2] C. A. Boano, F. J. Oppermann, K. Römer, J. Brown, U. Roedig, C. Keppitiyagama, and T. Voigt, "D-4.2 - Prototype of Testbeds with Realistic Environmental Effects," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., Oct. 2013.

[3] C. A. Boano, H. Wennerström, M. Zúñiga, J. Brown, C. Keppitiyagama, F. J. Oppermann, U. Roedig, L.-Å. Nordén, T. Voigt, and K. Römer, "Hot Packets: A systematic evaluation of the effect of temperature on low power wireless transceivers," in *Proceedings of the* 5^{th} *Extreme Conference on Communication (ExtremeCom)*, Aug. 2013, pp. 7–12.

[4] C. A. Boano, K. Römer, and N. Tsiftes, "Mitigating the adverse effects of temperature on low-power wireless protocols," in *Proceedings of the* 11^{th} *International Conference on Mobile Ad hoc and Sensor Systems (MASS)*. IEEE, Oct. 2014.

[5] C. A. Boano, M. Zúñiga, J. Brown, U. Roedig, C. Keppitiyagama, and K. Römer, "TempLab: A testbed infrastructure to study the impact of temperature on wireless sensor networks," in *Proceedings of the* 13^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2014, pp. 95–106.

[6] J. Brown, I. E. Bagci, U. Roedig, M. A. Zúñiga, C. A. Boano, N. Tsiftes, K. Römer, T. Voigt, and K. Langendoen, "D-1.2 - Report on Learning Models Parameters," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., Nov. 2013.

[7] J. Brown, J. Vidler, I. E. Bagci, U. Roedig, C. A. Boano, F. J. Oppermann, M. Baunach, K. Römer, M. A. Zuniga, F. Aslam, and K. Langendoen, "D-1.3 - Report on Runtime Assurance," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., Nov. 2014.

[8] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Swedish Institute of Computer Science, Tech. Rep. T2011:13, Dec. 2011.

[9] A. Dunkels, F. Österlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Sydney, Australia, Nov. 2007.

[10] A. Dunkels, "Rime — a lightweight layered communication stack for sensor networks," in *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*, Delft, The Netherlands, Jan. 2007. [Online]. Available: http://dunkels.com/adam/dunkels07rime.pdf

[11] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," in *Proceedings of the Conference on Local Computer Networks (IEEE LCN)*, 2004.

[12] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, "TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks," in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN'06)*, 2006.

[13] International Telecommunication Union, *Collection of the Basic Texts of the International Telecommunication Union adopted by the Plenipotentiary Conference*, 2011.

[14] LoRa Alliance, "LoRaWAN for developers," http://lora-alliance.org/For-Developers/LoRaWANDevelopers, 2014.

[15] F. J. Oppermann, C. A. Boano, M. Baunach, K. Römer, F. Aslam, M. Zúñiga, I. Protonotarios, K. Langendoen, N. Finne, N. Tsiftes, and T. Voigt, "D-3.1 – report on protocol selection, parameterization, and runtime adaptation," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., Jan. 2015.

[16] M. Sha, G. Hackmann, and C. Lu, "Energy-efficient low power listening for wireless sensor networks in noisy environments," in *Proceedings of the* 12^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2013, pp. 277–288.

[17] SigFox, "About SigFox," http://www.sigfox.com/en/about, 2014.

[18] N. Tsiftes, T. Voigt, F. Aslam, I. Protonotarios, M. A. Zúñiga, K. Langendoen, C. A. Boano, F. J. Oppermann, K. Römer, M. Baunach, J. Brown, U. Roedig, P. M. Montero, R. S. Hernández, M. Montón, and J. C. Pacho, "D-4.3 - First Integrated Prototype and Experiment," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., May 2014.

[19] Weightless SIG, "Weightless," http://www.weightless.org/, 2014.

[20] H. Wennerström, F. Hermans, O. Rensfelt, C. Rohner, and L.-A. Nordén, "A long-term study of correlations between meteorological conditions and 802.15.4 link performance," in *Proceedings of the* 10^{th} *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Jun. 2013, pp. 221–229.

[21] Worldsensing S.L., "BitCarrier Product page," http://www.worldsensing.com/our-solutions/ws-traffic-en/bitcarrier.html, 2014.

[22] M. A. Zúñiga, C. A. Boano, J. Brown, C. Keppitiyagama, F. J. Oppermann, P. Alcock, N. Tsiftes, U. Roedig, K. Römer, T. Voigt, and K. Langendoen, "D-1.1 - Report on Environmental and Platform Models," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., Jun. 2013.

[23] M. A. Zúñiga, F. Aslam, I. Protonotoarios, K. Langendoen, C. A. Boano, K. Römer, J. Brown, U. Roedig, N. Tsiftes, and T. Voigt, "D-2.1 - Report on Optimized and Newly Designed Protocols," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., May 2014.

[24] M. A. Zúñiga, I. Protonotoarios, S. Li, K. Langendoen, C. A. Boano, F. J. Oppermann, K. Römer, J. Brown, U. Roedig, L. Mottola, and T. Voigt, "D-2.2 & D-2.3 - Report on Protocol Models & Validation and Verification," http://www.relyonit.eu/, RELYonIT: Research by Experimentation for Dependability on the Internet of Things, Grant Agreement no: 317826, Tech. Rep., Nov. 2014.