Design of the Communication and Control Systems for Robotic Cleaning and Inspection of Solar Power Plants

by
Qianjun Tang

A thesis submitted
In partial fulfilment of the requirements
For the Master of Applied Science in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

© Qianjun Tang, Ottawa, Canada, 2021

Abstract

The aim of this research is to design the communications monitoring and control functionalities of an energy-efficient, scalable system, capable of supporting robotic cleaning and fault detection of photovoltaic panels, deployed in solar electric power generation plants. The communication functionality is implemented by using a wireless sensor network (WSN) deployed over the photovoltaic energy production plant's area. The network is designed to support the communication needs of static-sensing nodes as well as moving robotic units. It transports sensing data and commands between end units and the monitoring and control entity of the electric energy generation plant. Having robotic units replace humans in the cleaning and inspection tasks not only reduces the operational cost of the plant, but also results in increased energy production. Several innovations were necessary to achieve our objective, which are presented in this dissertation. A working prototype of the cleaning robotic system was built and tested in a solar power plant for a duration of 6 month The prototyping was done in collaboration of Tipot technology.

Acknowledgements

I would like to take this opportunity to extend my gratitude and thankfulness to my supervisor, Professor Dimitrios Makrakis, and my co-supervisor, Dr. Zhipeng Wang. During my study period, many important events happened in my family, including my father's death. Without their advice and review of my work, as well as their support and encouragement, it would have been difficult for me to complete my thesis.

I am also very much appreciative of the help I received from TIPoT Technologies Inc., which provided me the equipment and test field needed to complete my research.

Finally, I would like to express my gratitude to all of my friends and family members, whose help and support made my walk through this process easier and successful.

Contents

List of Figures

List of Tables

List of Symbols

C_{BF}: The capacity of the onboard battery of the fixed location node

C_{BM}: The capacity of the onboard battery of the robot

I_{FM}: The average current of the fixed location routing and sensing node's micro controller

I_M: The average current of the robot's micro controller

Imax: The maximum time interval between two DIOs

Imin: The minimum time interval between two DIOs

I_{PS}: The current of the robot position sensors

I_{RA}: The average current of the robot motors

L_T: The total traveling distance of the robot

P_F: The total power of the fixed location routing and sensing node

P_{FM}: The average power of the fixed location routing and sensing node's micro controller

P_{RA}: The average instant power of the robot motors

P_S: The instant power the onboard solar panel can produce

T_C: The time the solar panel can produce rated power per day

T_T: The total traveling time of the robot

V_B: The voltage of the onboard battery of the robot

V_{BF}: The voltage of the onboard battery of the fixed location node

V_{BS}: The voltage of the battery system per cell

V_M: The voltage of the robot's micro controller

v_R: The speed of the robot

V_{PS}: The voltage of the robot position sensors

V_R: The voltage of the robot motor system

W_{BF}: The total energy the onboard battery of the fixed location node

W_{BM}: The total energy the onboard battery of the robot needs to carry

W_M: The total energy needed for the robot's micro controller per day

W_{PS}: The total energy needed for the robot position sensors

W_R: The total energy needed for the robot motors

W_S: The energy the onboard solar panel can produce per day

W_{TM}: The total energy needed for the whole robot

W_{TS}: The total energy needed for the robot temperature sensors

x: The DIO clock doubling time.

Abbreviations

3G: Third generation of wireless mobile telecommunications technology

6LoWPAN: IPv6 over low-power wireless personal area network

ADC: Analog to digital convertor

AH: Amp hour

BHA: Bearing housing assembly

CAN: Controller area network

CGNPG: China General Nuclear Power Group

DAG: Directed acyclic graph

DAO: Destination advertisement object

DIO: DODAG information object

DIS: DODAG information solicitation

DODAG: Destination oriented directed acyclic graph

Estop: Emergency stop

ETX: Expected transmission count

GHz: Gigahertz

GPIO: General purpose input/output

GPRS: General packet radio services

GW: Gigawatt

IC: Integrated circuit

IDE: Integrated development environment

IEA: International Energy Agency

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force

IIC: Inter-integrated circuit bus

IoT: Internet of things

LAN: Local area network

LIFE: Lithium iron phosphate battery

LPM: Low power mode

MAC: Media access layer

MCU: Micro control unit

MHz: Megahertz

MPPT: Maximum power point tracking

MRHOF: Minimum rank hybrid objective function

MW: Megawatt

NUD: Neighbor unreachable detection

O&M: Operation and maintenance

OF: Objective function

OS: Operation system

P2P: Peer to peer

PCB: Printed circuit board

PHY: Physical layer

PLR: Packet loss rate

POE: Power over ethernet

PV: Photovoltaic

PWM: Pules width modulation

RF: Radio frequency

RLT: Route lifetime

RoI: Return on investment

RPL: Routing protocol for low-power and lossy networks

RSSI: Received signal strength indicator

SCADA: Supervisory control and data acquisition

SPI: Serial peripheral interface

SRCIS: Smart robotic cleaning and inspection system

TCP: Transmission control protocol

UDGM: Unit disk graph medium

UDP: User datagram protocol

VDC: Voltage for direct current

WAN: Wide area network

WH: Watt hour

WIFI: Wireless fidelity

WSN: Wireless sensor network

μIP: micro-IP

Chapter 1

Introduction

1.1 Background of Solar Power Plants

Use of solar energy for the production of electric power is a highly promising technology for generating environmentally friendly (“green”) energy [39]. Judging from lessons learned over the past several years by studying deployments of solar electrical power generation plants, it is a financially sound technology that should be included in the development of sustainable power-generation infrastructure [39]. In 2019, approximately 131 TWh of electric power were generated by the existing solar photovoltaic (PV) infrastructure [39], which corresponds to a 22% increase in the yearly global solar PV production during 2018. This boosted the global solar PV generation capacity to 720 TWh, as per the International Energy Agency (IEA) [39]. Figure 1.1 illustrates the solar PV capacity from year 2000 to now and the prediction of the development of solar PV capacity from 2025 to 2030 [39]. Exponential growth is evident in the production volume of solar energy, and it could reach capacity at the range of 3,268 TWh by 2030. Figure 1.2 illustrates the trend for each renewable energy development 2000. It can be seen that solar PV generation has become more and more important in the renewable energy

category.

Solar energy is clean, but it is not free. The installation of a solar electric power generation plant costs a significant amount of money, depending on the site's location and size. For this reason, the specific location of a solar electric power generation plant must be selected carefully. Most of the large-scale solar electric power generation plants are located in remote, open areas lacking co-location of major obstructions that could block sunlight. Another important consideration is that solar electric power generation plants in low-altitude areas are more productive, as the sunlight falls almost perpendicularly on the panels and there is less solar elevation angle change through the year, which ensures that the power production is rather even. For maximum efficiency, many power plants are located near the equator where there is strong direct sunlight exposure all year long. Additionally, unless the panels are equipped with automatic tracking capability of sun's location, the solar panels need to be installed with a small inclination angle or be placed flat to better face the sunlight.

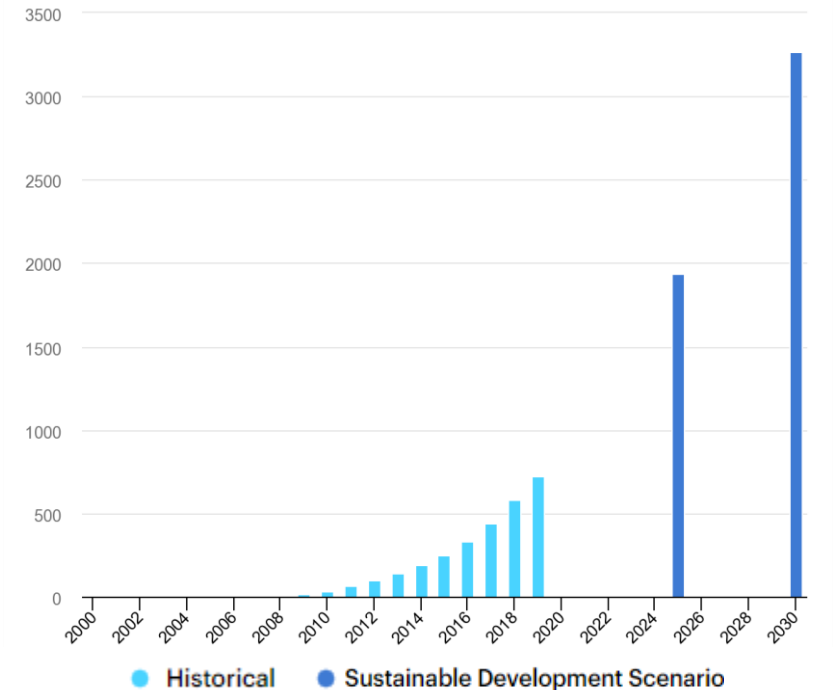


FIGURE 1.1 Solar power generation for the period of 2000 to 2030. (In TWh) [40]

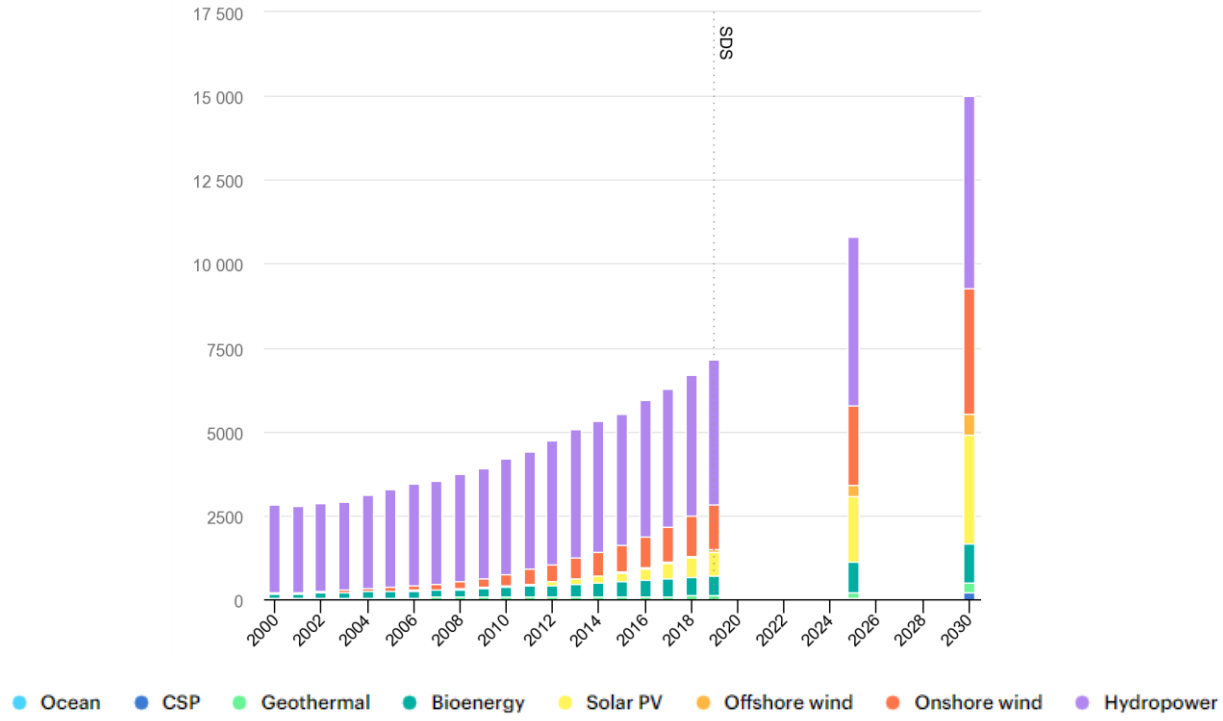


FIGURE 1.2 Renewable power generation growth by technology for the period 2000 to 2030. (In TWh) [41]

1.2 Detection of Soiling Loss and Solar Panel Defects

Most solar electric power generation plants, particularly the large-scale ones, are facing the following major challenges in their daily operation and maintenance (O&M).

1. **Soiling losses**

   Soiling on a solar panel—such as bird droppings, lichen growth, dust clouds, and dirt buildup—can cause significant power generation loss [42]. Solar electric power generation plants tend to be located in areas where soiling levels are often heavy. Because most plants are built in remote places that lack trees or buildings (to maintain a high level of sun exposure), with some of them even deployed in deserts, the impact

of storms and the soiling level associated with them is high. Figure 1.3 illustrates the optimal installation angle of solar panels at various altitudes. The smaller the installation angle, the higher the accumulation of soil or sand on the surface of the panel, because the soiling material will not fall to the ground. The higher the accumulation of soiling materials, the lower the efficiency of the solar panel becomes. Thus, especially in sites near the equator, soiling of the panel's surface is a serious problem. Figure 1.4 illustrates an example of dust accumulation on solar panels at a solar electric power generation plant at the Hashemite University in Zarqa, Jordan. The left side of the figure depicts the solar panels with dust accumulation, while the right side of the figure illustrates them after they have been cleaned. As pointed out in [42], the sun energy to electrical energy conversion rate can be reduced by more than 20% due to soiling. Rain does not solve the problem, since the mix of water and soil produces mud, which sticks and solidifies on the surface if not physically removed. Since large-scale solar electric power generation plants usually extend over a wide geographic area, the cost of solar panel cleaning is quite significant. Currently, most cleaning is performed manually, which is labor-intensive and costly, making it difficult to be performed frequently. Usually the operator can afford to clean a site once every yearly quarter or even less frequently. As displayed in Figure 1.5, since solar panels are connected serially, the production imbalance generated by even a single dusty panel can cause drop of productivity to the remainder of the connected panels.

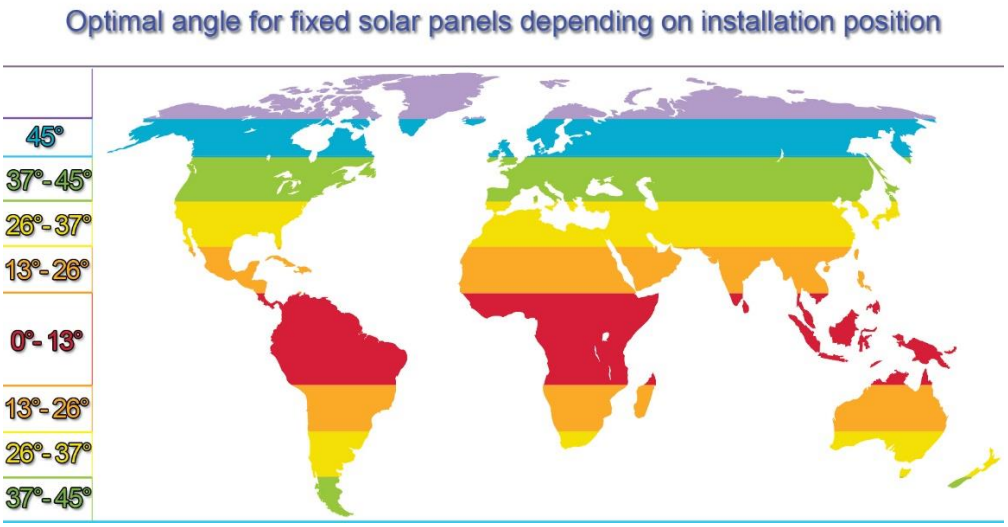


FIGURE 1.3 Optimal installation angle for solar panels at different altitudes. [43]

FIGURE 1.4 Dust accumulation on solar panels of an operational solar electric power generation plant. [44]

To the mentioned challenges, the shortage of water in many locations where electric power generation plants are deployed should be added. This has led to the development of waterless cleaning technology (e.g. [45], [46]).

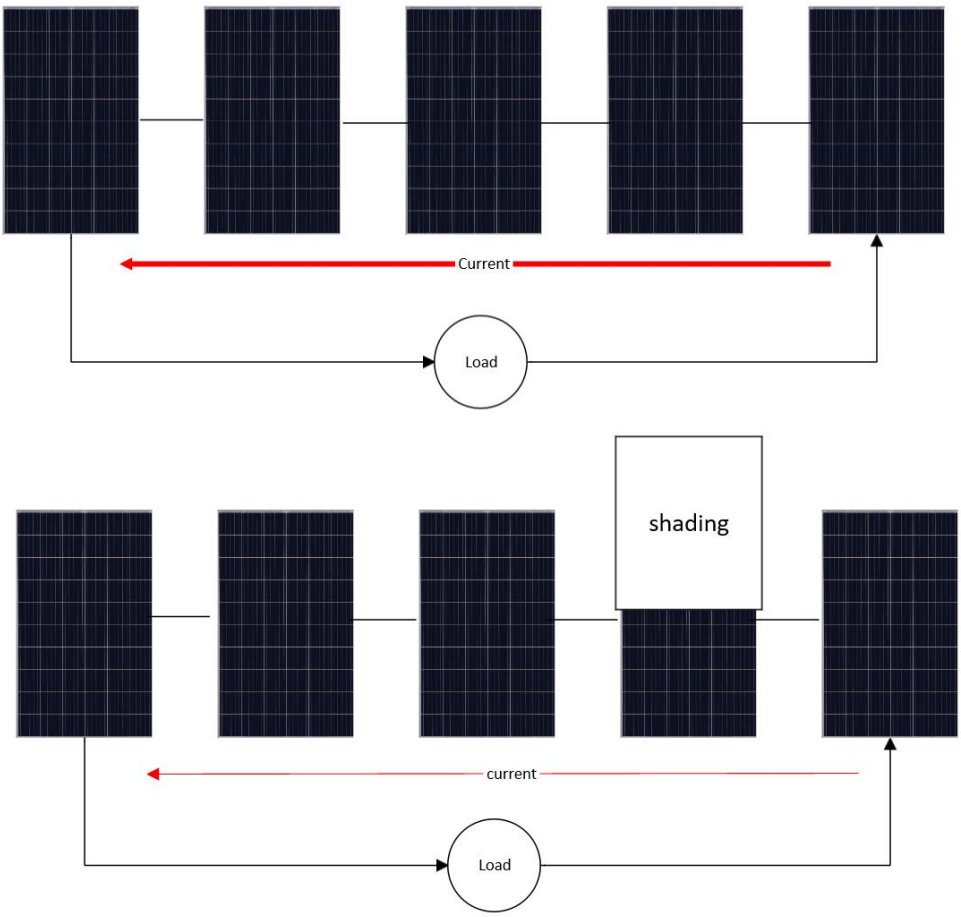


FIGURE 1.5 Serially connected solar panels affect each other's production efficiency.

Currently, several companies, such as [45], [46], [47], and [48], are developing cleaning robots to deal with the photovoltaic panels' soiling problem. There are some noticeable weaknesses in these products. Most of these cleaning robots use timer or infrared-based technology for remote control. Some of the robots use general packet radio services (GPRS) or WiFi for communications. GPRS service is not available in remote sites where there is no cellular network coverage, and WiFi is not suitable for large-scale plants because of its coverage area limitations. Additionally, to the best of our knowledge, none of these robots have the ability to detect and cross obstacles on the solar panel rows, such as a bearing housing assembly (BHA) and gear box. Figure 1.6 illustrates the BHA obstacle. The housing circled in red protrudes from the solar

panel surface, disrupting the path of the cleaning brush.

2. Solar panel defects detection

FIGURE 1.6 Bearing housing assembly at a solar electric power generation plant.

Defects in solar panels—such as micro-cracks, shading (which cannot be removed even with daily cleaning and is also considered to be a defect), and uneven soldering—pose a frequent and complicated challenge for efficient O&M. They are also one of the main causes of malfunctioning, even rendering panels inactive. The detection of solar panel failure can be as simple as finding “hot spots” [49] existing on a solar panel in a solar array row. Hot spots occur when a panel is shaded, damaged or electrically mismatched. When the panel encounters these problems, it loses part of or all of its ability to convert solar power to electricity. However, the panel is still exposed to the sunlight, and heat will accumulate in the surface of the panel, causing its temperature to rise [50]. Figure 1.7 displays the thermal image of several panel rows. The relation between color and temperature is at the middle of the thermal image.

Just by visual inspection, one can easily identify the defective solar panels. The values appearing below in the form of tables display the instant electricity production in watts for each panel, so one can see that panels with hot spots do not produce any power.

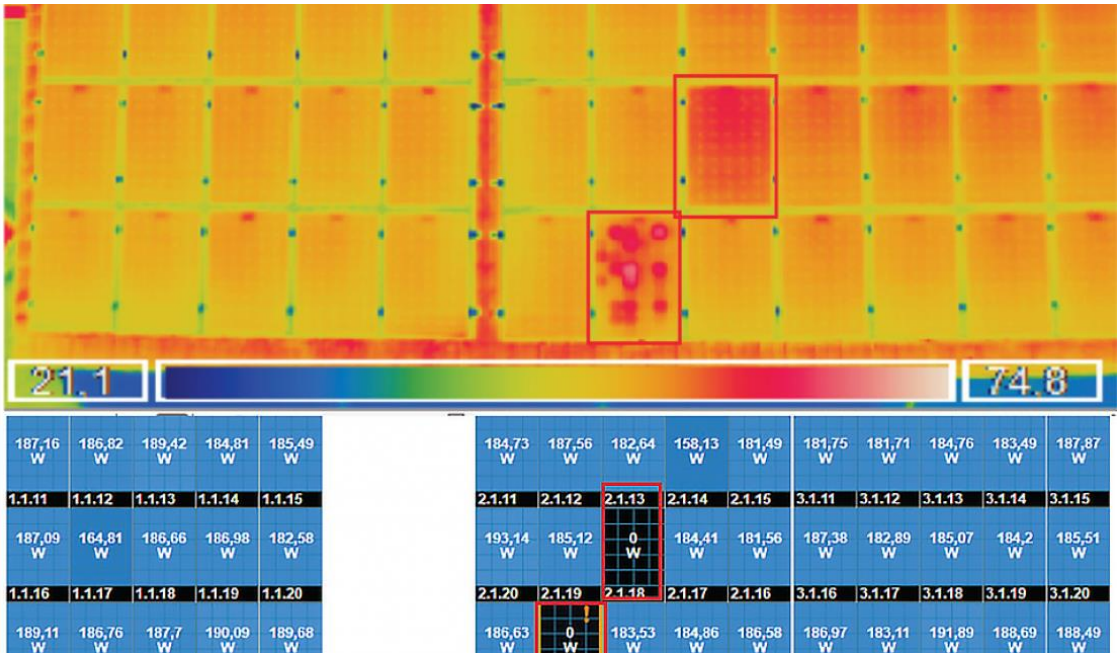


FIGURE 1.7 Example of “hot spots” in a solar array row as they appear in a thermal image. [51]

Figure 1.8 indicates that shading can also cause hot spots on solar panels. The left side of the figure depicts a solar panel with dust accumulated at its bottom edge, while the right side depicts a thermal photograph of the same solar panel. It can be seen that the temperature of the shaded area is higher compared to the other areas of the panel.

FIGURE 1.8 Shading causes “hot spot” on solar panel. [52]

Currently, detection of defects on solar panels is performed manually, which is highly inefficient and labor-intensive. Some companies are using drones with thermal cameras to perform hot-spot detection [53], as illustrated in Figure 1.9. Expensive equipment and trained drone operators are required for the job. For large-size plants, scanning the entire site is a lengthy process, and the accuracy of the results can be affected because of the distance between the drone and the PV panels. Additionally, weather conditions affect the effectiveness of the method, since winds could prevent or hamper the operation of the drone, as well as produce variability of distance from the panel due to wind-forced drifts.

FIGURE 1.9 Drone with thermal camera performing “hot spot” inspection. [53]

1.3 Motivation of Research

Statistical evidence indicates that solar electric power generation plants are becoming a major component of the energy system, with solar power generation expected to increase over the next decade [39]. A solar electric power generation plant, regardless of its location, needs maintenance such as cleaning and health monitoring. Currently, most solar electric power generation plants rely on manual cleaning and inspection, which is not only inefficient but also costly. For typical solar electric power generation plants of MW or GW capacity, there are hundreds of solar rows between several hundred meters and several kilometers long. If the cleaning cannot take place at a reasonable frequency, dust accumulation and animal waste deposits will solidify over time and become quite difficult to remove. Additionally, if solar panel defects cannot be detected in a timely fashion, the power plant could suffer low electricity generation efficiency and other solar panels could be damaged.

With the rapid development of large-scale solar electric power generation plants, solutions that can effectively lower O&M costs and improve a plant’s performance become more and

more important to this industry. Particularly, there is a strong and rapidly growing demand for autonomous and reliable solar panel maintenance systems, which aim to significantly reduce labor costs for solar panel cleaning and, in the meantime, detect defects or potential risks with higher operational frequency. Such systems will offer significant benefits to the solar electric power generation plant's owners and operators, for example, through system performance improvement, lifetime extension of solar panels, reduction of operation costs, and maximization of the return on investment (RoI) of solar projects.

1.4 Objectives of Research

A smart robotic cleaning and inspection system (SRCIS) normally consists of cleaning and inspection robots, a wireless communication network, a server with a software management system, and a client application or user interface. The robots provide autonomous cleaning and hot-spot inspection of solar panels. These tasks should be flexibly scheduled by the solar electric power generation plant operator. The collected sensor data should be transmitted to the server, and the robots can receive control commands from the server. Additionally, the robot should be self-sustainable so that the operation of the system is independent of the power plant. In this thesis research, our objective is to design and implement the communication and control subsystems of the SRCIS, and investigate the effectiveness of wireless sensor network (WSN) technology in providing wireless communication between moving robots and the server.

The robots move along the solar rows and brush off the dust lying on the solar panels. As they do so, they can be used to check the temperature of the solar panel while working to detect any hot spots. The control module governs the movement of the robot as it performs the

cleaning job. It is also responsible for collecting sensor data and managing power. The control module works with the communication module to send data to the server and receive commands.

As mentioned earlier, a large-scale solar electric power generation plant covers a large area. Use of network nodes transmitting at high power levels is inefficient because they are energy-limited. This conflicts with the need to have such network nodes be self-sufficient energy-wise; they must recharge themselves through their own small solar panels. Thus, a multi-hop wireless communication network approach is more suitable for covering a wide area in an energy-efficient manner. Therefore, area coverage, energy limitation and data rate is the main concerns for this application. Such a network will transport sensor data and control commands. To be internet-compatible, the nodes run on the transmission control protocol/internet protocol (TCP/IP) stack [22]. Since the robots are moving when performing cleaning and inspection tasks, efficient and reliable support of node mobility by the WSN is necessary. Furthermore, the design must take into account constraints such as weight and system cost.

In summary, the objectives of this thesis research are to:

- Design and implement the control module of the smart cleaning and inspection robot and test the functionalities.
- Test the improvement of energy generation for robotic cleaning.
- Design and implement the WSN for the transmission of sensor data and control commands between robots and server.
- Evaluate the ability of the designed WSN to support mobile sensing nodes through comprehensive performance evaluation.

- Propose mechanisms that improve the network performance in terms of transporting sensor data and control commands.
- Ensure that the design is both cost-effective and energy-efficient.

It is worth of pointing out that the actual hotspot detection algorithm is not in the scope of this thesis. This thesis is focused on the research and implementation of the smart robotic cleaning system and its supporting network. It is capable of sensing and transmitting needed data for hotspot detection. The detection algorism needs to be implemented on the cloud server as an export system.

1.5 Contributions

The major contributions of the research reported in this dissertation are the following:

1. Design and implementation of a robotic control system.
    a. Design of the control system architecture. (With the collaboration of Tipot Tech.)
    b. Design of the self-charging power and power management module.
    c. Development of the firmware for collecting data from various sensors.
    d. Development of the firmware that implements control logic and drives motors and the servomechanism (servo). (With the collaboration of Tipot Tech.)
    e. Development of the firmware for transmitting sensor data and receiving control commands. (With the collaboration of Tipot Tech.)
    f. Development of the firmware for different nodes as per their needs.
    g. Completion of a field functionality test for evaluating the effectiveness of the

robotic cleaning. (With the collaboration of Tipot Tech.)

2. Design of a supporting network meeting the needs of the application and evaluation.
    a. Study of the deployment strategy of WSN nodes.
    b. Evaluation of the impact of gateway location on network performance, including network topology, energy consumption, and packet loss rate through simulation.
    c. Performance of a simulation to gauge the benchmark performance for a routing protocol for low power and lossy networks (RPL)-based WSN when supporting mobile sensing nodes.
    d. Proposal of several mechanisms to improve the performance of the WSN and achievement of reliable data transmission for mobile sensing nodes in robotic cleaning and inspection.
    e. Completion of extensive computer simulations to verify the effectiveness of the proposed performance improvement mechanisms.

1.6 Organization of the Thesis

The remainder of the thesis is organized into five chapters. A description of the content of each chapter is provided here.

Chapter 2 presents the relevant background technologies and reveals how the SRCIS could be designed and implemented with Internet of Things (IoT) technologies. This chapter provides an overview of some of the key IoT technologies, such as WSN and embedded operating systems. Additionally, this chapter reviews some of the existing problems in the RPL layer of

the WSN, evaluates the performance for a moving node in a multi-hop WSN, and compares various modifications of the RPL protocol.

Chapter 3 describes the related hardware and software used in this thesis research. Hardware includes WSN nodes, sensors used to collect information, and actuators that mechanically implement the issued commands. Software includes three parts. The first is the software that is needed to develop the firmware for the WSN nodes, the second is the embedded operating system that runs in the micro-controller unit (MCU) of the WSN nodes, and the third is a WSN simulator used when designing and testing the RPL design for mobile sensing nodes.

Chapter 4 presents the design and implementation of the communication and control subsystem for the SRCIS. Key aspects of the design are discussed in detail, including subsystem architecture, power management module, control logic, and communication network nodes. A field functionality test is also conducted to verify the design and implementation, as well as the effectiveness of the robotic cleaning.

Chapter 5 focuses on the performance of the RPL protocol for gateway location variation and supporting mobile sensing nodes, since the robots in the SRCIS are moving along the solar panel array to perform the tasks. Based on the evaluation results, several strategies were devised and developed that enhance the performance of the WSN in supporting mobile sensing nodes. Three different mechanisms are proposed, and the improved performance is validated through extensive simulation.

Chapter 6 concludes the thesis and discusses some possible future directions. Applications that have moving nodes within a WSN and their related RPL modifications are also discussed in this chapter.

Chapter 2

Technology and Literature Review

2.1 IoT Overview

IoT is the new paradigm of the internet. If the concept of the internet is to connect as many computers as possible, the concept of IoT is to connect as many objects as possible. Such objects could be sensors, actuators, microphones, and cameras. Data can be transferred to the cloud for processing, decisions could be made, and messages could be conveyed to actuators to implement action. A typical IoT application can be divided into three parts. At the bottom lie the sensors and actuators, which generate data and control the device. The middle houses the transmission part, which transports information from sensor to the cloud server and vice versa, by delivering command, control, and operation messages to actuators and devices. On top is the cloud server, where collected data are processed and analyzed and the control logic that generates the issued control commands is applied.

IoT is a multi-discipline technology, which covers electronics, the embedded system, communication technology, networking, cloud computing, web design, and control algorisms. Each discipline contributes to the IoT application in a different way. Here, we focus on the

communication technology and how it applies to IoT. Currently, WiFi and Bluetooth are the most popular wireless communication technologies, however, they both have a large energy footprint. In relation to IoT technologies, they might be prime candidates for some applications, but if restrictions for energy consumption are present, other technologies might have an edge.

2.2 WSN and Its Implementation

A WSN is designed to collect and transport sensor data and to control small actuators in a distributed fashion. WSNs have found application in a multitude of services, including environmental monitoring. Thousands of temperature sensors can be deployed in a forest to monitor for presence of fire, and motion and location sensors can be placed to monitor water flow or wind flow, which could provide advanced warning of a developing flood or formation of tornados [1]. For modem city surroundings, stain gauges and rotation sensors can be used to monitor the “health” of buildings, bridges, and tunnels. For applications closely related to our daily lives, temperature and light intensity in a room as well as humidity and carbon dioxide concentration levels inside a large building can be monitored, and collected data can be used constructively. Appropriate processing and analysis of these valuable applications have been made possible by the use of distributed sensor networks, which are quite different from other wireless communication networks, such as WiFi, Bluetooth, and a 3G/4G cellular network, in terms of data rate, power consumption, and so on. A WSN is designed for these special needs and it is intended to connect more and more things to the internet, in line with the concept of IoT [54]. With IoT, people can create a better living environment, investigate the physical world in greater depth, and increase productivity. These are only a few of a growing multitude of

areas that can benefit from the IoT.

Parameters critical to the applicability and performance of WSN devices include size, power consumption, resilience, mobility, and scalability [2]. Most WSN devices need to be of small size for convenient deployment in various places [3][4]. The small sensor nodes are usually battery-powered. It is important that a WSN sensor node has a small power consumption footprint. In most of the use cases, people want a sensor node that is “invisible” so as not to interfere with their daily lives. For example, people want to know whether a door is opened for security reasons, but they do not want the sensor to be broadly visible. This is not simply for aesthetics but also to prevent would-be thieves from spotting the sensor nodes and adjusting their plan for a break-in by developing strategies to tamper with them physically or electronically. Physical size limits the device’s computation capacity, memory capacity, energy storage, and the sensors it can support. The communication protocols running within the device are significant consumers of processing power, memory, and energy. Thus, with the size limitation comes the need for the network protocols running in the nodes of a WSN to be simple, so as not to consume a significant amount of energy (see [5] for a detailed explanation). On top of the mentioned requirements, there is the issue of cost per node, which should remain low. One more factor affecting the proliferation of WSNs through our society is the level of ease of deploying and maintaining the nodes and the level of their self-sufficiency regarding formation, maintenance, and lifespan of the WSN. Users wish to have the same level of ease they currently enjoy with deploying a WiFi-based home or small business network. Achieving these objectives is quite a challenge, especially when dealing with WSNs involving node population at the scale of thousands that cover a large geographic area where easy access and mobility is

challenging.

In most cases, a WSN node has a highly limited transmission range because of the energy constraints associated with it, which does not allow transmission at high power levels. All of the sensor nodes need to send back a data packet to a gateway node that connects to the internet [11]. In many application scenarios, a WSN should cover a large geographic area where some sensor nodes are placed far away from the sink/gateway, thus there exists a conflict between low transmission power and large-scale coverage. Multi-hop is then introduced to a WSN to solve this problem. In a WSN, sensor nodes can pass the collected information to their neighbors, which then route the information hop-by-hop to the sink. This feature achieves a satisfactory balance between transmission power and coverage but requires that, should an intermediate node fail, the WSN be capable of establishing an alternative routing path from the specific sensor node to the sink [14]. Otherwise, the node's data, along with the data of all other nodes having this node as part of their per-hop-path to the sink, will be lost.

In several scenarios of WSN use, nodes remain at the same location, forming a static topology. However, there are cases where the nodes are located on moving/mobile objects, thus the topology changes with time. In such a case, the WSN needs to be flexible and adaptable [13].

An example is military tactical networks, with the nodes located on the individual soldiers and the military vehicles engaged in combat. Similar scenarios exist with devices located on police officers or firefighters. In such a case, maintenance of a connected WSN becomes challenging. Additionally, there is a growing need for deployment of a larger number of

interconnected sensor nodes, which creates the challenge of WSN scalability. Significant advancements have been made through the years to address the challenges, yet there is still need for more innovation in the area of a WSN.

Concerning the structure of a "mainstream" WSN: The node's structure can be segmented into five modules: power module, computing module, radio transceiver module, sensor module, and actuator module [6].

The functionality of the power module is obvious. In a medium- to small-size WSN node, a power module consists of a battery pack and some voltage regulators, so that the system can have the correct input voltage for each integrated circuit (IC) contained in the node's electric circuitry [6].

The computing module is the core unit of the WSN device; it processes the collected data and runs the communication protocol stack. Normally, a 16-bit or 32-bit MCU is used for the implementation of this module. Based on the complexity of the task, the computing unit can be built to run a micro–operating system or a regular communication [11].

The radio transceiver module is used to transmit radio signals to other WSN nodes of the network and receive them as well. The most popular commercial WSNs operate at the 2.4GHz unlicensed band allocation. Lower-frequency bands such as 700 MHz to 900 MHz are used for achieving longer-range transmission [12].

The sensor module contains the sensors that are attached on the WSN node. Two types of sensors are used to collect data from the physical world: analog and digital. Analog sensors produce an analog signal in a form of voltage or current level. The signal is then sampled by

the microcontroller as it performs an analog to digital conversion (ADC) to generate a digitized version of it, suitable for inclusion in a packet. Digital sensors are analog sensors in their core but with added ICs to complete the ADC internally. Digital sensors pass the digital signal to the MCU through the node's internal digital communication bus. The number of analog sensors that can be connected to a sensor node is usually limited by the number of ADC ports on the MCU, while the number of digital sensors is limited by the protocol of the communication bus the microcontroller and sensor uses [12].

If need be, actuators can be connected to the WSN to realize designed functionalities. Most actuators can be controlled with general purpose input and output (GPIO) pins on the MCU. Some more advanced actuators must be interfaced with such communication protocols as serial peripheral interface (SPI) [91] and inter-integrated circuit bus (IIC) [92].

2.3 6LoWPAN in WSN

A WSN extends the connectivity of the internet through its connection to a gateway node. Each sensor node, in addition to communicating with its peers or the root, might need to send data to or be reachable from networked nodes located outside the WSN. Such nodes could be personal computers, smartphones, tablets, and so on, which wish to gain access to the collected information or pass control instructions to certain nodes [16]. TCP/IP is the standard networking protocol suite for the internet and it is widely used around the world. The 128-bit IPv6 address space can support the allocation of individual IP address in the range of 3.4×10^{38},

which enables the support of countless smart devices (including sensor nodes), each having its own IP address [17]. In addition, IPv6 with its built-in security module and the ability to include "custom made" headers make it more reliable and flexible than IPv4.

The full TCP/IP protocol suite is too heavy for WSN nodes, which have limited power, memory, and computation resources. A 6LoWPAN (IPv6 low-power wireless personal area network) [55] is an adaptation layer developed to enable IPv6 functionalities running in Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 [56] based devices. IEEE 802.15.4 is the IEEE standard for a WSN. Figure 2.1 [10] provides a comparison of the 6LoWPAN stack to other standardized network stacks. As mentioned earlier, 6LoWPAN is an adaptation layer, which connects the media access layer (MAC) layer and IP layer to enable IPv6 over a WSN.

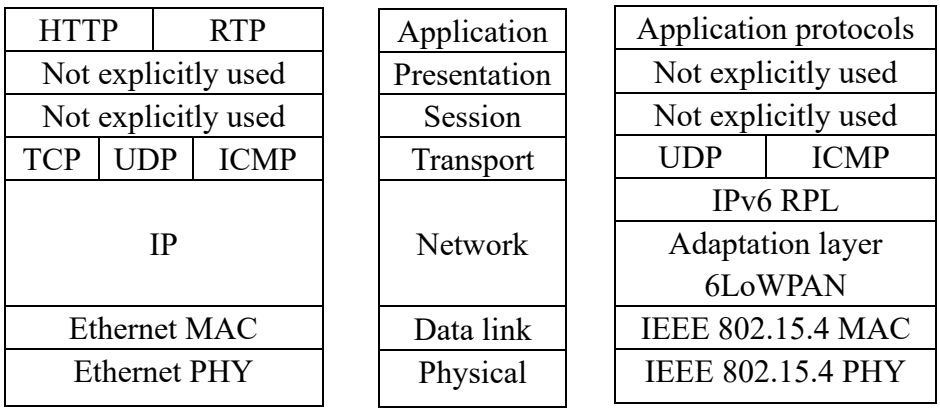


FIGURE 2.1 The TCP/IP, OSI, and 6LoWPAN stack. [57]

A micro-operation system (OS) or embedded OS is normally used for WSN nodes running a 6LoWPAN. Most micro-OS implementations have the 6LoWPAN network stack built in and allow the developers to speed up their research and development (R&D) process. Some popular embedded OSs include TinyOS [8], ContikiOS [7], and Riot [9].

2.4 The Routing Protocol for Low-Power and Lossy Networks

The purpose of routing is to find a path that can deliver datagrams from source to destination. In a WSN, three different types of traffic—namely, node-to-root traffic, root-to-node traffic, and node-to-node traffic—might exist, which the implemented routing protocol must move through the network and deliver to the destination.

RPL was developed to solve the routing problem for a WSN over a 6LoWPAN [19]. The main challenge is that the routing process must be carried out with highly limited resources (e.g., memory, computation power, bandwidth) and while operating in a lossy environment. As stated in Section 2.1, the routing protocol for a WSN must be simple, resilient, and scalable. A destination-oriented directed acyclic graph (DODAG) [19] is used to implement the RPL, which can cope with the WSN characteristics quite well. DODAG is a directional graph, which illustrates the routes available for each node to reach the root [18]. The concept of an RPL is simple: It uses a ranking system to identify the hop-distance from a sensor node to the root. A sensor node always forwards the datagram to a lower-ranked node, known as its parent, until the rank hits zero, which corresponds to the root. Clearly, a sensor node only needs to keep information of its parent for its up-link traffic (traffic going to the root). To better understand an RPL, we need to review three important concepts, namely, the DODAG, RPL control massage, and the trickle timer algorithm, then conduct a review of the RPL routing formation process.

2.4.1 Destination-Oriented Directed Acyclic Graph (DODAG)

An RPL forms a DODAG routing topology. DAG [32] is a tree-like topology with no

loops and DODAG is a special form of DAG, where the topology has only one sink node that corresponds to the root. An example of a DODAG topology is illustrated in Figure 2. Node 2 is the root of the DODAG.

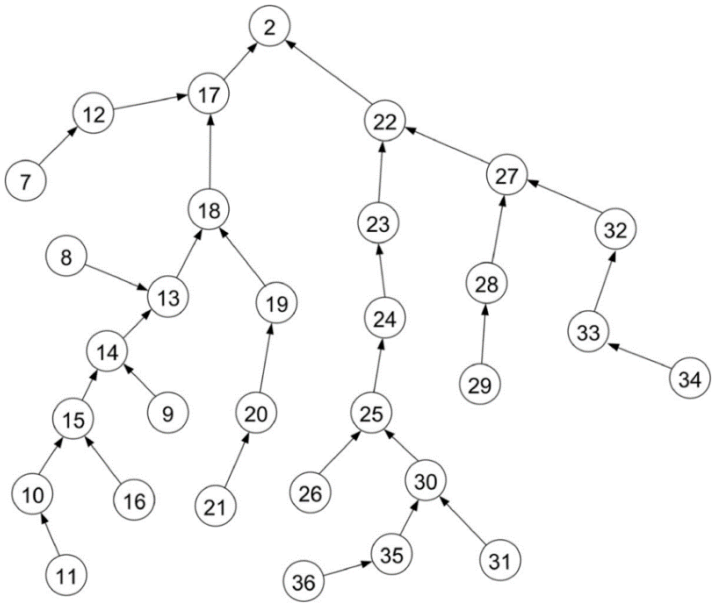


FIGURE 2.2 DODAG. [58]

2.4.2 RPL Control Messages

There are four control messages in the RPL protocol: (i) DODAG information object (DIO), (ii) DODAG information solicitation (DIS), (iii) destination advertisement object (DAO), and (iv) DAO-Ack [19]. A brief introduction description of each of them is provided below, with more detailed explanation provided in Section 2.4.4.

1. DIO is a multicast control message that carries DODAG information. DIO is used to advertise the existence of a DODAG to new nodes (and they can decide if they wish to join the DODAG) or allow member nodes to check the consistency of the DODAG.
2. DIS is used by a node that is not a part of a DODAG to request a DIO from its surrounding nodes that is already part of a DODAG.

3. A DAO is a request message from the new node to the potential parent node, asking to join DODAG. This request is generated after this new node processed the DIO it has received and decides to join the DODAG.
4. DAO-Ack is generated by a parent node, or the root. It is the answer for any DAO request, and it can be a “yes” or a “no.” A yes will allow the potential child node to become part of the DODAG. A no will force the node to look for new DODAG.

The above control messages of an RPL are used to form the DODAG of a WSN.

2.4.3 The trickle timer algorithm

The trickle timer algorithm [24] is used to control the DIO schedule. An RPL utilizes a dynamic scheduling system to control the time intervals between two DIOs. The logic behind that is to reduce the volume of control messages while maintaining the WSN functioning as per expectations. When the network is forming or when there is inconsistency in the network, more DIO messages are needed to quickly adapt to the changes. When the network is stable and consistent, fewer control messages are needed [24]. For example, if the network is stable, we can set the minimum and maximum DIO interval to be 4s and 128s. The starting DIO interval for the node is 4s; after 4s it increases exponentially until it reaches 128s, where it remains. If the network is not stable, upon sensing the change of the network (a DIO from a new node or a parent node rank change and so on), the node will revert to its minimum DIO interval and repeat this process.

2.4.4 RPL Routing Formation

Two additional concepts, the objective function (OF) and the rank, should be discussed

before undertaking explanation of the RPL routing formation process. The OF is a set of rules or constraints that a node uses to calculate its own rank [31]. For different applications, different OFs can be developed, based on the need of the application. The most common OF is based on hop-distance, which is named “of0” in the Contiki system [19], and expected transmission count (ETX) [60], which is named “mrhof” and stands for minimum rank hybrid objective function, in the Contiki system [59]. Both hop-distance and ETX are routing metrics. Hop-distance describes how many hops a packet, generated by the node, goes through to reach the root. ETX is the average number of transmissions made for a packet to be received without error at its destination [59]. They both can be used to evaluate the cost and quality of a route. Other routing metrics can be used to develop other OFs as well, such as latency, received signal strength indicator (RSSI), and node battery energy level. The rank of a node is calculated through the routing metric value, computed by the routing algorithm that combines and processes the relevant OF. The calculated routing metric value is included in the DIO and is used in the formation of DODAG [19]. DIO is an advertising message that contains the rank information of the node that sends it out. The DODAG forming process starts as a rank assignment for each sensor node [27]. A simple description of the procedures is presented in Figure 2.3:

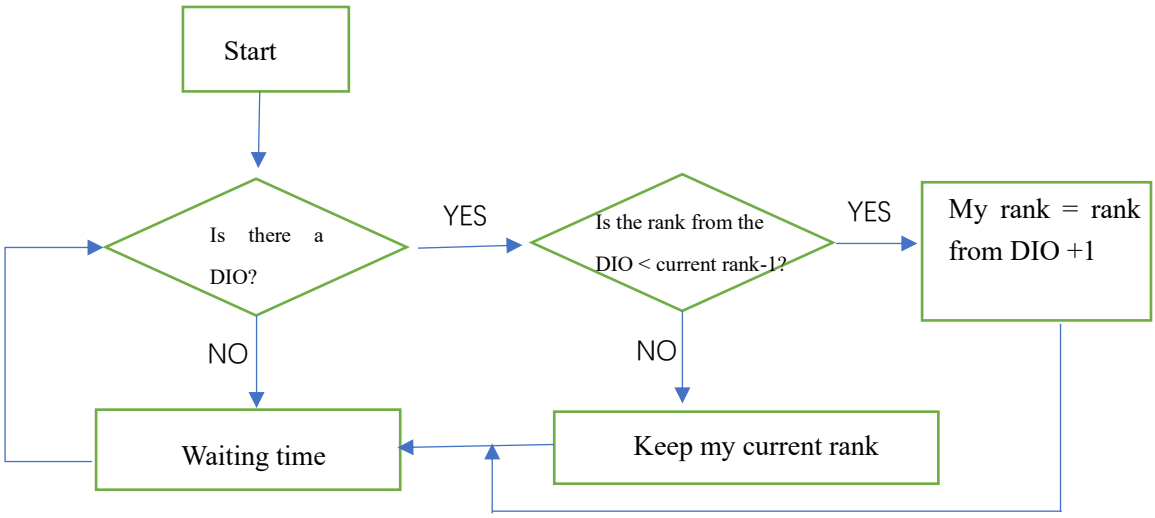


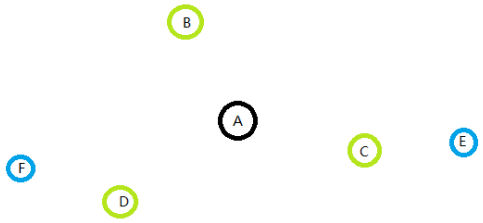
FIGURE 2.3 DODAG forming process.

1. After activating, the sensor node will listen for a DIO advertisement.
2. When the sensor node receives a DIO advertisement, it retrieves the rank information contained in the DIO massage, denoted here as *N*, and compares it with its current rank. If the sensor node does not have a rank, or its current rank is greater than *N*+1, it will rank itself *N*+1.
3. When a sensor node calculates its own rank, it will start to advertise it.
4. This process continues until all sensor nodes become ranked
5. The process will continue even after every node in the network is ranked, so that new sensor nodes can be connected into the network.

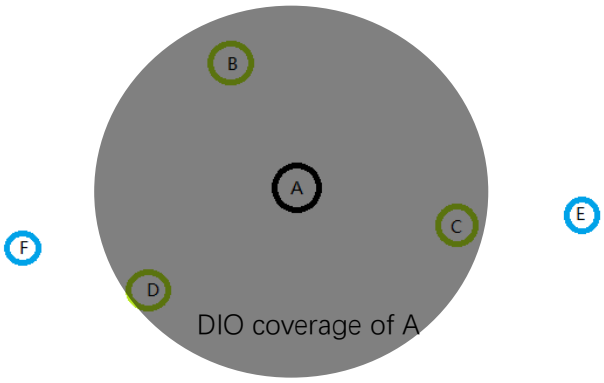
After a node has calculated its own rank, it will store the sensor node, which advertised the DIO massage leading to the determination of the rank, as its parent; therefore, the DODAG is beginning to form. The DODAG formation process is explained below through an example. As illustrated in Figure 2.4, the DODAG formation process starts from the root node A. After initialization, A begins to broadcast DIO massages, each of which contains two important

pieces of information, namely, node identity and rank. In the coverage area of A, there are sensor nodes B and C. Upon receiving the DIO from A, they notice that A ranks 0, the lowest rank in the network, thus they will rank themselves as 1 and store A as their parent. Then they begin to send out their own DIOs to advertise themselves as well.

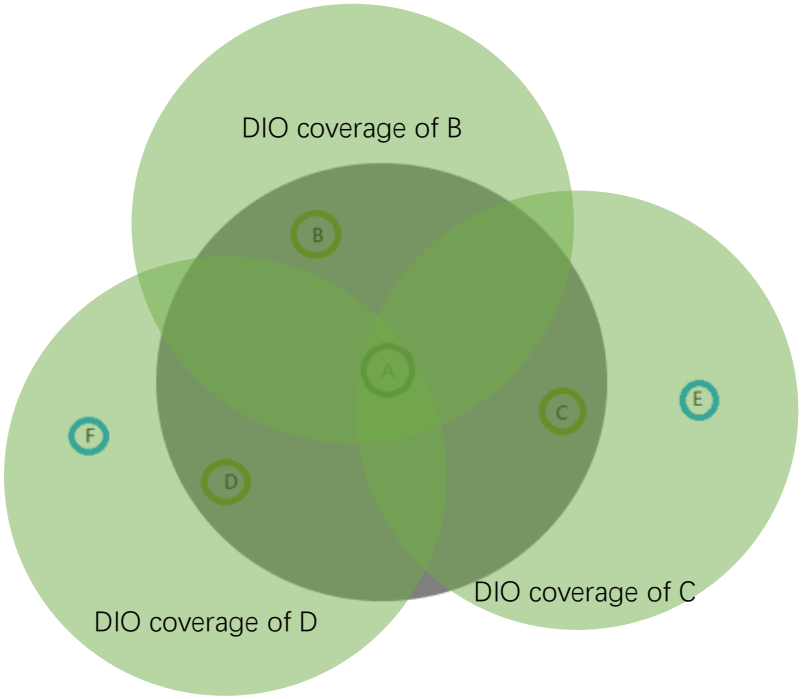
A graphical representation of this process can be seen in Figure 2.4 and Figure 2.5.



(a) Sensor nodes before the DODAG formation



(b) The DIO coverage of the root node A



(c) the DIO coverage of nodes with rank 1

FIGURE 2.4 DODAG formation.

TABLE 2.1 Rank and parents stored in each node.

Sensor node	Rank	Parent
A	0	N/A
B	1	A
C	1	A
D	1	A
E	2	C
F	2	D

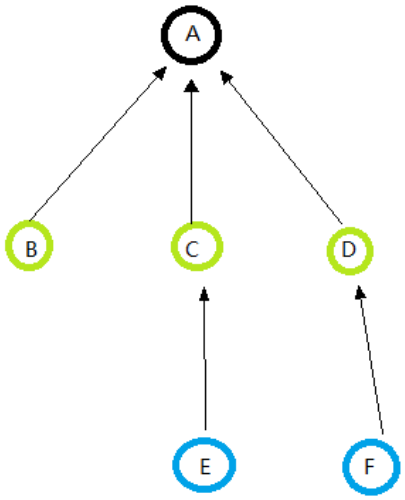


FIGURE 2.5 DODAG topology.

With the DODAG being formed, the uplink for sensor data flow (from child node to the root) is established, which means all the sensor nodes can now send data to the root. The downlink, where the data is sent from root to sensor nodes, is somewhat different. There are two modes for the downlink: the storing mode and the non-storing mode. In the storing mode, every sensor node keeps a routing table for all sensor nodes having higher rank that use this node as a hop to get connected to the root. In the non-storing mode, only the root keeps a routing table to all the sensor nodes. The routing table is produced by the destination advertisement object (DAO) message. When a DODAG is formed, all the end nodes (or so-called “leaves”) begin to send a DAO message to the root. This message contains information about end node reachability and the path to get there, in this case, all the intermediate nodes through which the DAO message travels on its way to the root. Continuing with the example of DODAG formation presented above, Table 2.2 indicates a complete routing table for the

storing mode.

TABLE 2.2 Rank, parents, and routing table stored in each node.

Sensor node	Rank	Parent	Destination
A	0	N/A	B, C, C/E, D, D/F
B	1	A	N/A
C	1	A	E
D	1	A	F
E	2	C	N/A
F	2	D	N/A

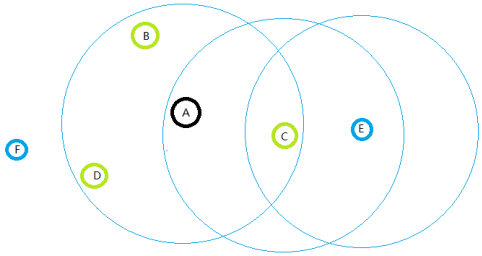
Although the RPL protocol meets the requirements of a WSN, it has its weaknesses and problems. For example, RPL works well for uplink data collection, but for the downlink data transmissions (e.g., when some control instructions need to be sent to a sensor node), the root must store the complete routing table to send the instructions to a sensor node. The ranking system limits the mobility of a sensor node, as it only allows a node to change from a higher rank to a lower rank [18]. These issues are discussed in detail in Section 2.5.

2.5 Issues with the RPL Protocol

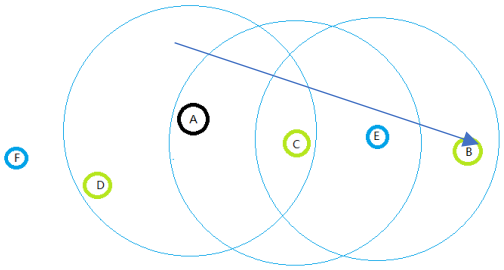
We focus mainly on two issues that the current RPL protocol has as they relate to this thesis: the rank restriction issue and the neighbor unreachable issue. Both of them impact negatively support of mobility by an RPL-based WSN.

1. Rank restriction: From the DODAG forming process, it can be noticed that a

node can only lower its rank. There is no mechanism to make a node change its rank from a lower number to a higher one. This would not be an issue if all of the sensor nodes had fixed locations. However, this could become a problem for mobile nodes, as the node can move away from the root in terms of hop-distance. If a node moves away from the root and out of its original parent's coverage, it will be unable to maintain its network connection even if it is in the cover area of another sensor node due to this rank restriction [19]. The node will eventually be treated as an unreachable node and it will take a long time for this node to reconnect to the network. It will reconnect only after a "global repair" takes place, in other words, when the network runs again the formation of DODAG, thus refreshing the routing topology. Figure 2.6 displays a scenario where rank restriction could force loss of a sensor node's network connectivity due to mobility. In the figure, the mobile node is node B. In the new location of node B, reconnection cannot be established quickly due to rank restriction and unawareness of the movement. Node B was in the coverage area of the root node A, and thus had rank 1. After moving to the new location, it is in the coverage of node E, which has rank 2, as it is the child of node C. Node B cannot immediately take node E as its new parent because of the rank restriction. Instead, it must wait for a global repair process.



(a) The initial topology of the WSN

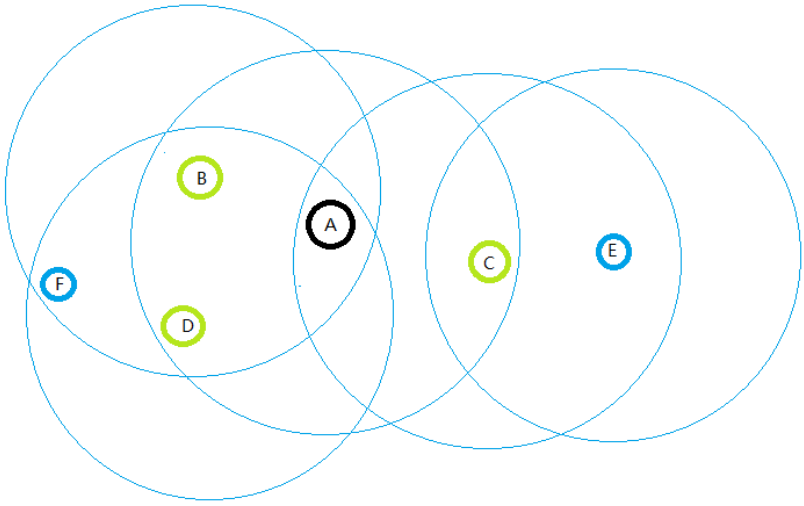


(b) Node B moved to a new location

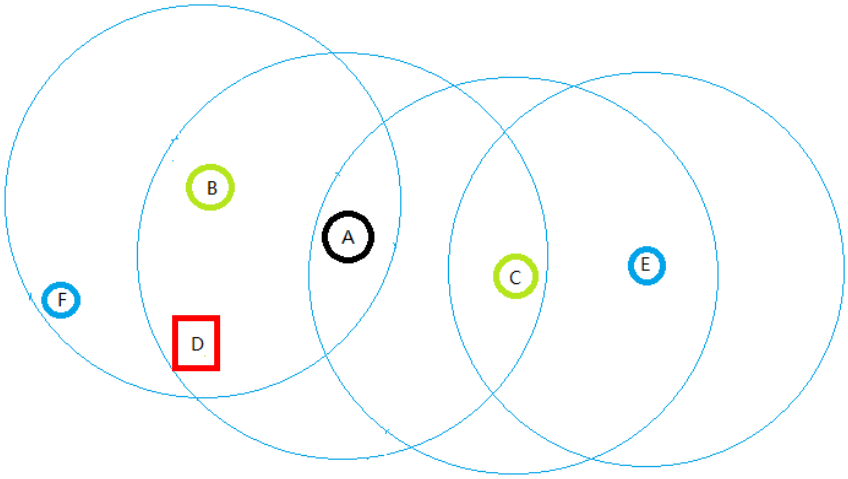
FIGURE 2.6 Mobility and rank restriction scenario.

2. Neighbor unreachable: There is no neighbor unreachable detection (NUD) mechanism in an RPL protocol [34]. It should be pointed out that there are several reasons for having a neighbor unreachable, for example, location change of the node, node failure, and node traffic congestion. Figure 2.7 illustrates the neighbor unreachable case. When parent node D for node F becomes disconnected from the network, it will take a long time for sensor node F to switch to a different parent (node B in Figure 2.7b) and re-establish full connection to the network, due to the following reasons. When there is no NUD, the root cannot know there is a link breakage until it wants to send a datagram through that node and learns that the datagram cannot be delivered, through feedback information sent from the MAC layer [96]. The root becomes aware of the link breakage and responds to it. On the

mote side, the node uses the DIO to check the network consistency. Specific settings will be determined by the RPL configuration file, but the DIO period is not fixed. To reduce control overhead, the DIO period could be set from several minutes to hours. Therefore, the update for network consistence might experience a significant delay, which could lead to a long reconnection time [21] [24]. This period of disconnection will cause sensor information loss, not only for this sensor node, but for all nodes that use this node as an intermediate hop in their routing path to the root.



(a) Node F uses a parent node D to root A.



(b) Node D failed and caused a neighbor unreachable problem.

FIGURE 2.7 Neighbor unreachable scenario.

2.6 Related Research Contributions on Accommodating Mobile Nodes in WSNs Implementing RPL.

As defined in the RPL protocol of a 6LoWPAN, the rank of a sensor node can only change from a higher number to a lower number, which allows a sensor node to move only toward the root in terms of hop distance. When having this mixed setup of both static and dynamic sensor nodes, a new one-bit flag in the DAO message is suggested by Cobarzan et al. [29] as an indication of a mobile sensor node. The reverse trickle algorithm was proposed in [29]. This algorithm is based on the assumptions that a mobile child node is constantly moving with a fixed speed and is likely to stay for a while after it is first connected to a parent node. As time progresses, the probability of the node having moved out of the coverage area of the parent node increases. Upon receiving the mobile node's DAO request and accepting it as its child, the parent node will switch to the reverse trickle algorithm, thereby decreasing the interval of

its DIO advertising time. The decision about what the value should be depends on the used RPL configuration setting as established by the node. These settings should be selected in accordance with the node's coverage area, the speed of the mobile node, and the data rate of the mobile node [100]. For example, if the mobile node is moving quite slow, the WSN is relatively stable, thus a longer maximum DIO interval can be used to reduce control overhead. If the mobile node is moving quickly, a shorter maximum DIO interval can be used to increase the update rate. By doing so, the mobile child node can have a higher frequency of checking the network consistency when the probability of it having left the coverage of its parent node is higher. As mentioned earlier, the assumption is that the longer a mobile node stays in its current parent's coverage, the more likely it becomes that the mobile node will go out of the coverage of this parent. Based on the reverse trickle timer algorithm, the longer a mobile node stays in its current parent node's coverage, the more often the parent node will advertise DIOs for the mobile node to check the network consistency. A timer is set up on the mobile node to record the inter-arrival time of DIO messages; then, based on the running average of this timer, the mobile node updates a threshold value every time it receives a new DIO. Another timer is used to determine whether the new DIO arrives on time (within the threshold). As soon as the mobile sensing node concludes it cannot confirm the reachability of the parent node, it will reset its own rank to infinity and begin the network formation process locally. When using the reverse trickle algorithm, the parent node requires that a mobile child node send a DAO every time it receives a DIO, in order to make the parent aware of its presence. Once the parent node verifies there is no mobile sensing node in its coverage area, it will switch its DIO schedule to normal mode. This method is based on the assumption that the mobile sensing node will remain

in its parent's coverage long enough. The duration a mobile node should remain within the coverage of the parent to allow the assumption to hold can be calculated as follows: We use the radius of the parent node's coverage circle (half of the longest linear path that a mobile node can have in a parent node's coverage) to represent the average distance that the mobile node must travel to move out of the signal range of the parent node. This distance should be divided by the speed of the mobile node, and this time must be at least 1.5 times the maximum DIO interval. Note that DIO can only be expressed as a power of 2 in a WSN network. If the stay time is too short, the mobile node will leave the coverage of the parent node before the reverse trickle algorithm can even respond. This method prompts the mobile node to transmit more DAOs to confirm its existence with the parent node. More DAO messages lead to more power consumption for the mobile node, which already has a tight power budget, due to limited space for battery capacity and power consumption from the motor; they are all powered by the same battery. The extra power consumption due to the transmission of more DAO messages negatively impacts the energy efficiency and potentially the operational effectiveness of the mobile robotic node.

Another technique, RSSI-based coverage awareness mRPL, is proposed by Fotouhi et al. [30]. The mobile child node utilizes the RSSI reading from the received DIO messages generated by its parent to estimate its location change relative to its parent node. By comparing the average RSSI and the set threshold, the mobile node will decide if it needs to look for new parent. If the mobile node decides it needs a new parent, it will multicast a DIS message and wait for responding DIOs. Based on the RSSI readings of the received DIOs, the mobile node will make its decision and will select as its new parent the one corresponding to a DIO with the highest

RSSI reading [28]. However, an accurate location estimation depends on the reliability of the RSSI reading. The RSSI can be used as a satisfactory reference to estimate location, but it is environment-dependent. It might work well when the channel is in prime condition and there is no significant interference, however, these are not conditions to be expected very often in WSN deployments. RPL, as its name suggests, has been designed for and is used on low-power devices and a lossy environment. As mentioned earlier, the accuracy of RSSI readings becomes questionable under such conditions. RSSI is used as a routing metric for the OF, to calculate rank. If the rank calculation is made using only the RSSI, the accuracy of the calculated rank becomes questionable. It may cause the mobile node to make wrong decisions, such as leaving the old parent node too early or choosing a parent that is far from optimal. Frequent switching of the parent node might be the result; this could also lead to an oscillatory behavior (a mobile node keeps switching between two possible parent nodes).

From the above discussion, we have realized four important facts. First, this change-of-parent process performed by a mobile node consists of two phases—the leaving phase and the reconnecting phase. Second, it might be beneficial to the functionality and stability of the WSN that mobile sensing nodes always remain leaf nodes (meaning they do not carry any child nodes). This can be implemented by canceling the DIO schedule of all mobile sensing nodes (preventing them from sending DIO messages). Without advertising themselves, no node will connect to any of them as a child node. Not only does this contribute positively to the stability of the WSN, it also reduces energy consumption of mobile nodes, and there is no additional energy cost associated with a mobile sensing node moving away from its location and rejoining the network at another location. Third, the DIO interval is an important parameter to the

functionality of mobile nodes within the network. A DIO transmission setup might not be as important when mobile nodes are not present within the WSN, however, in the presence of mobile nodes, a DIO interval can affect the update rate of certain information and cause a mobile node or its parent node to react slowly. Last, a hybrid OF, utilizing more than RSSI parameters in the routing metric, might help the mobile node choose the right parent node.

Chapter 3

Hardware and Software Platform

3.1 Hardware

This section focuses on hardware that forms the IoT platform with an emphasis on the WSN framework. The hardware that is responsible for the WSN can be divided into three parts—the communication and control unit, the sensors, and the actuators. Most of the WSN has a highly integrated main controller, which includes a power module, an MCU, and a radio frequency (RF) module. The MCU is the core unit that operates the communication stack, takes care of the networking protocols, and manages devices connected with the WSN node. The RF module takes care of the physical layer of the communication function. This combination forms a WSN node. Sensors and actuators are then connected to the node through various interfaces to form a complete sensing node.

The WSN hardware for the solar power generation plant must consider the following aspects. First, it must be able to handle multi-tasking, so an embedded operating system is preferred. Second, the sensing nodes must have a large enough coverage range so the number of fixed-location routing and sensing nodes required to cover the service deployment area can

be kept low. The test site[1] used eight fixed-location routing and sensing nodes to cover a solar array row 1 km long. Each node is placed at a distance of 130 m from its neighboring nodes. Third, a variety of sensors must be placed on the node, so a platform that can support easy, cost-efficient integration of sensors of diverse nature is required. In our design, a total of seven types of sensors are attached to the sensing nodes—two types of temperature sensors[2], one wind speed sensor, one wind direction sensor, one voltage sensor, one current sensor, and one proximity sensor. Most are analog sensors, therefore requiring an ADC interface. Some digital sensors need IIC support. Third, all nodes have a limited power supply, as the required energy is produced by (relatively small) photovoltaic panels connected to the nodes. A balance between performance and energy consumption is necessary. Lastly, we are already familiar with the programming apparatus required for development and simulation. The choice for the WSN hardware testing platform is multifarious.

Table 3.1, taken from [61], lists most existing testing platforms. Eleven out of the twelve platforms listed in the table use a micro-operation system to help manage multi-tasking, but only six of those 11 are running on IEEE802.15.4, which is the industry standard for a WSN. We chose to follow this standard, since the IEEE802.15.4 regulated hardware is more suitable for WSN applications in terms of energy consumption and RF signal regulations. The requirement for sensor connectivity support shortens the list further, leaving only two competitors among the six candidates. The iMote2 platform and the TIPoT (tiny IP of things)

[1] Located in Jiangshan, Zhejiang, China, it was used to test the functionality of the robot; all of the parameters and setups of the system are based on the requirements of this testing site.

[2] One analog based temperature sensor for operation temperature reading, and one digital based contactless temperature sensor for hop-spot detection.

platform comply with the IEEE 802.15.4 standard, and each has the sensor interface to support all sensors that will be connected to the sensing node. The TIPoT platform uses Contiki OS, an open-source micro controller OS, which is built on top of the 6LoWPAN network stack. It can be deployed with limited resources such as energy, memory, and computation power and can handle complicated applications [7]. Additionally, TinyOS, in the iMote2 platform, is an open-source micro controller OS, but it is focused mainly on wireless communication and sensor data collection. It has difficulty handling complicated applications [8]. In our application, the micro controller's OS needs to handle control applications in addition to sensing, data collection, and communication, therefore, Contiki OS is more suitable for the requirement of this thesis. The TIPoT platform has a signal coverage range of 180 m under the line of sight; when deployed in a solar electrical power generation plant, its range will be affected by the presence of solar panels, but it is still capable of providing more than 150 meters of coverage.[3] Based on these considerations, we decided to use the TIPoT WSN development kit and liner of products as our hardware platform. The specifications of TIPoT's [62] WSN development kit are provided in Table 3.2.

[3] Field test to verify the range of signal coverage by TIPoT was carried out in October 2017.

TABLE 3.1 WSN hardware platform and specifications summary. [61]

Node	Operating System	CPU	Memory Flash	Wireless Interface	Interfaces	Sensors
iSense	iSense	32bit(16MHz) ARM	128KB (96KB)	IEEE802.15.4 (2.4GHz)	USB, RS-232	Temperature, Light, PIR, AMR, Accelerometer
iMote2	TinyOS	32-bit (13-416MHz) Intel PXA271	32MB (32MB)	IEEE802.15.4 CC2420 (2.4GHz)	USB, JTAGI2C, SPI, UART, GPIO	Temperature, Accelerometer, Humidity, Light
Scatter Node MSB430	ScatterWeb	16-bit(4MHz) MSP 430F1612	55KB (5KB)	CC1020 (868MHz)	JTAG, GPIO	Temperature, Humidity, Accelerometer
MicaZ	Contiki	8-bit (8 MHz) Atmega 128L	128KB (4KB)	IEEE802.15.4 CC2420 (2.4GHz)	JTAG, USB	Temperature, Light, Microphone, Accelerometer
Tmote Sky	TinyOS	16-bit(8MHz) MSP430F1611	48KB (10KB)	IEEE802.15.4 CC2420 (2.4GHz)	USB	Temperature, Light, Humidity
ESB	ScatterWeb	16-bit(8MHz) MSP430F149	60KB (2KB)	TR1001 (868.35MHz)	RS2-32, JTAG	Temperature, Light, Noise, Vibration, Infrared
Sun SPOT	None (JVM)	32-bit (180MHz) ARM920T	4MB (512KB)	IEEE802.15.4 CC2420 (2.4GHz)	USB, GPIO	Temperature, Light, Accelerometer
Scatter Net MSBA2	Contiki	32-bit (72MHz) ARM7	512KB (98KB)	Chipcon CC1100 (864-970MHz)	miniUSB2.0, GPIO	Temperature, Humidity
GumStIx	Linux	32-bit (400MHz) Intel PXA255	16MB (64MB)	IEEE802.11b/g (2.4GHz)	USB, Bluetooth	Water Depth
Pacemates	iSense	32-bit (60MHz) LPC2136	256KB (64KB)	Xemnics RF (868MHz)	Serial, I2C, Radio interface	Heart rate monitor
TelosBs	TinyOS/ Octopus	16-bit(8MHz) MSP430F1611	48KB (10KB)	IEEE802.15.4 CC2420 (2.4GHz)	USB	Temperature, Light, Humidity

TABLE 3.2 Specifications for TIPoT mote. [62]

Node	Operating System	CPU	Memory Flash, (RAM)	Wireless Interface	Interfaces	Sensors
Tipot	Contiki	STM32F103 32-bit (72MHz)	128KB	AT86RF231 IEEE802.15.4 2.4GHz	Mini USB Serial, I2C, SPI, GPIO, CAN	ADC to interface with any analog sensor

3.1.1 TIPoT IoT Development Kit

TIPoT's IoT development kit is an integrated WSN development environment. It includes mote nodes, a gateway node, and a sniffer, which captures wireless communication packets for analysis. Both the mote and gateway nodes are equipped with a stm32f103 MCU [64] as the main control unit and an RF module AT86RF231 [65], which is responsible for 2.4GHz 802.15.4 wireless communication. The Contiki [7] operation system can be installed on both mote and gateway nodes to support a 6LoWPAN communication protocol. Figure 3.1 depicts the content of a basic TIPoT IoT kit. It includes a gateway node, two mote nodes, a sniffer, and a programming adapter.

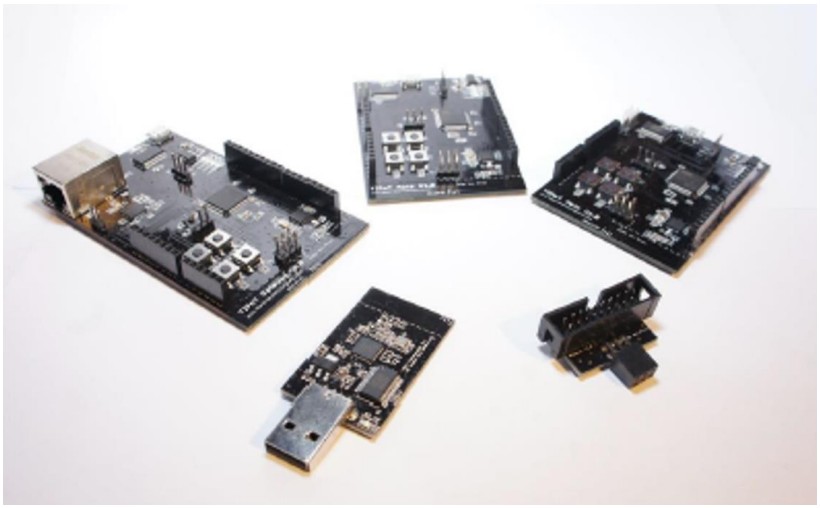


FIGURE 3.1 TIPoT development kit. [62]

3.1.1.1 TIPoT Mote

The TIPoT mote device is a basic unit of the TIPoT IoT development kit and is used to develop sensing nodes for WSN. It is a highly integrated node with a power module, control unit, and RF module, all built in one printed circuit board (PCB). Contiki OS is used to manage all devices integrated within the node and all the tasks to be carried out with the node. The primary task for a mote is to maintain the WSN connectivity; it will take care of the network formation and respond to any change that has been made inside the network. In addition to this primary task, the mote interfaces with sensors and actuators. A WSN node can be treated as a smart sensor or actuator with network connectivity capability. Figure 3.2 depicts a TIPoT mote. GPIO pins and special function pins, which are used to interface with other devices, are located on both sides of the board.

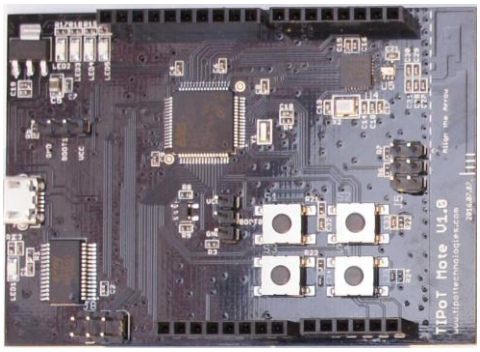


FIGURE 3.2 TIPoT mote [62].

3.1.1.2 TIPoT Gateway

The main difference in hardware between a TIPoT mote and a TIPoT gateway is that the gateway has one more module on top of the TIPoT mote, which in our current implementation is an ethernet module. A gateway interconnects two different types of networks. Depending on the resources and the requirement, different modules can be added to the gateway to achieve other connectivity possibilities. For instance, a WiFi module can be added if the application chooses to use WiFi as the access network to the internet. Another option is a cellular network,

in which case a 3G shield is installed on the gateway. Figure 3.3 depicts a TIPoT gateway; the added ethernet module and Rj45 port can be seen at the lower left corner of the board.

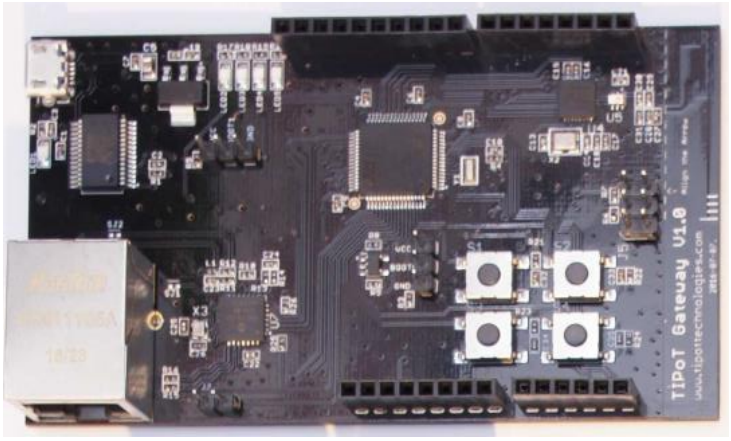


FIGURE 3.3 TIPoT gateway [62].

3.1.2 Sensors

Sensors can be analog or digital, with the differences based on the type of signal the sensor puts out. An analog sensor puts out an analog signal, expressed as a voltage or a current, having magnitude that is proportional to the size of the physical entity the sensor is measuring. To interface with an analog sensor, the main controller must have an ADC to transform it into digital form (bit pattern) that will be sent through our computer networking infrastructure. The resolution of the ADC is affected by the number of bits the ADC uses to express an analog value. Depending on the sensor type, a low pass filter might be used to interface the sensor and the ADC. It will help to eliminate high frequency noise and provide anti-aliasing. A digital sensor will put out a converted digital number through a digital communication interface instead. The ADC is done by the sensor itself. The main controller will need a digital interface to communicate with the sensor to receive the sensor readings. Popular communication protocols for this purpose are the SPI, IIC, and serial bus. Analog sensors provide raw, uncalibrated data, and processing is required to remove environmental imperfections or biasing

factors, the removal of which generates readings of higher quality. Digital sensors have done this calibration internally. A multitude of sensors have been used with our system, including a temperature sensor, wind speed and direction sensor, current and voltage sensor, and proximity sensor.

3.1.2.1 Temperature Sensor

Temperature is an important factor in the health monitoring of a solar electric power generation plant, and must be monitored at many different sites. A hot spot on a solar panel often indicates a bad connection or possible damage to that particular section. The earlier a hot spot is noticed, the less it will affect the remainder of the solar panel array. Overheated transformers or inverters could cause a fire, and they also must be detected as fast as possible to minimize loss. Temperature sensors can be contact or contactless. Most contact temperature sensors have an analog output, and we must interface them with the onboard ADC of the MCU. Most non-contact sensors are infrared-based and usually have a digital output interface. Figure 3.4 (a) depicts a DS18B20 contact temperature sensor [79], which has a measuring range from minus 55 to 110 degrees Celsius with a 5V analog output interface. Figure 3.4 (b) illustrates a GY-906-DCI non-contact IIC-based digital temperature sensor [80], which measures from minus-40 to 85 degrees Celsius within a distance range of 1 m.

The contact temperature sensors can be placed at fixed locations, frequently with power inverters and transformers where detection of abnormal temperatures caused by overloading or short-circuiting is necessary. A non-contact temperature sensor is installed at the edge of the cleaning and inspection robot to scan for hot spot on the solar panels during the detection process.

(a) contact [79]

(b) non-contact [80]

FIGURE 3.4 Temperature sensors [79] [80].

3.1.2.2 Wind Speed Sensor

Wind can produce energy, but it can also cause damage. Therefore, wind speed must be monitored to help determine whether the cleaning and inspection task should stop to allow the robot to be locked down on the frame for safety purposes. Due to its relatively light weight, the robot could be shifted in a strong wind; therefore, locking is needed to ensure that it is immobilized. Wind-speed sensors are placed at the boundary of the plant, and collected data enable the operator to decide in a timely fashion when the robot should be locked on the frame. Wind-speed sensors are normally analog. Figure 3.5 depicts a YGC-FS-5V-V2 wind-speed sensor [81], which takes measurements from 0 to 45 m/sec and has a 5V analog output interface.

FIGURE 3.5 Wind-speed sensor [81].

3.1.2.3 Wind Direction Sensor

To make an optimal assessment about whether to stop cleaning and monitoring operations and lock the robot on the frame, wind-direction details are required in addition to wind-speed

information. If wind blows from the back side of the moving robot, it can lift the robot from the solar panel array and drop it on the ground, with fatal consequences. Additionally, wind blowing opposite to the moving direction of the robot would add load to the robot's driving system or even stop the movement of the robot, causing electrical damage. Wind-direction sensors are normally analog sensors as well, with the wind direction being proportional to the output voltage. Initial calibration is required to determine the wind blow direction to which zero-voltage corresponds. Figure 3.6 depicts a YGC-FX-5V-V2 wind-direction sensor [82], which also takes measurements from 0 to 360 degrees and has a 5V analog output interface.

FIGURE 3.6 Wind-direction sensor [82].

3.1.2.4 Voltage Sensor

A solar electric power generation plant is designed to be a closed system, which means the functionality of the smart maintenance system is as independent as possible from the energy resources of the plant. This limits the maximum energy the cleaning and inspection robot can harvest each day. If the robot is not fully charged when starting its task, it may stop halfway through them on a solar array, which is an undesirable situation. The robot then would become a sunlight blocking entity, creating shadows on that solar panel section and having the potential to lead to a hot-spot problem, significantly degrading the transformation rate of the entire array.

Thus, a decision must be made prior to sending the cleaning and inspection robot to perform its task. The voltage level of a battery is a reliable indicator of a battery's capacity, with the voltage level proportional to its capacity within its designed working range. Figure 3.7 depicts an OPEN-SMART, voltage-divider-based voltage sensor [83], with a measurement range up to 25 VDC and a 5V analog output interface. The voltage sensor is installed on the robot to monitor the battery charging level. On top of that, the battery voltage level is also a useful indicator for battery health conditions, serving as a satisfactory parameter to assess the battery life cycle and develop an effective replacement schedule.

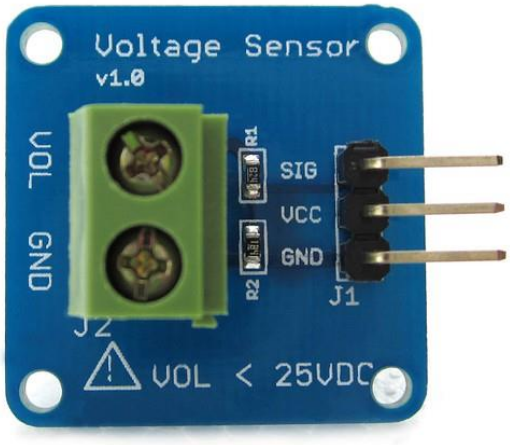


FIGURE 3.7 Voltage sensor [83].

3.1.2.5 Current Sensor

Current is an important parameter when one wishes to assess whether the operational condition of an electrical device is operating normally. Figure 3.8 depicts an ACS770LCB-100U-PFF-T current sensor [84], which is capable of measuring DC current up to 100A and has a 5V analog output interface.

Other than electrical-related problems, some mechanical failure can also be detected by monitoring the current change of a driving motor of a system. When the cleaning and inspection

robot becomes immobile due to an obstacle, with the robot continuing its effort to move, the current of its driving motor will increase in an attempt to overcome the extra workload. If an unexpected peak in the current value appears in the current-versus-time progression curve, there might be a problem, such as a misalignment of the solar panel array as a result of a solar panel shifts due to a loose bolt. Current can be also used to monitor the battery-charging process. Instead of using the voltage as an indicator, the input current can be monitored during the day, when the onboard battery of the cleaning and inspection robot is charging, and used to determine the battery capacity.

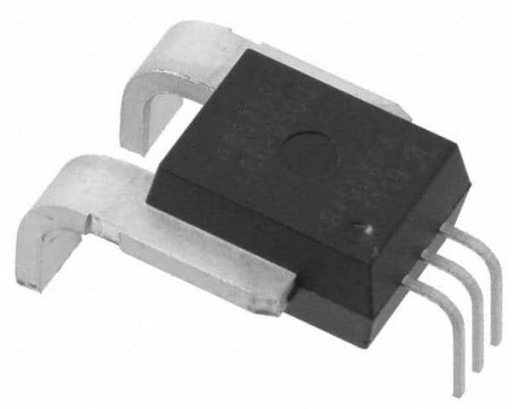
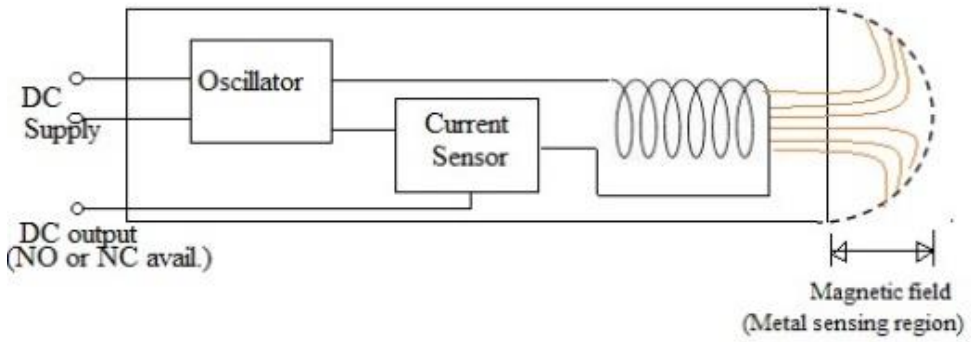


FIGURE 3.8 Current sensor [84].

3.1.2.6 Proximity Sensor

A proximity sensor detects whether any metal object is in its range. A main steam trigger distance for this type of sensor is from 2 mm to 20 mm, depending on the model. This sensor is equipped to gain position information for the cleaning and inspection robot. Metal markers are installed on the solar array at the docking station and the return station, allowing the proximity sensor to read them for robot position. Figure 3.9 (a) depicts an E2AM18LS08M1B1 proximity sensor [85]. It has a threaded body, which allows the user to adjust the installation location so as to control the triggering point. Figure 3.9 (b) depicts the internal structure of such a sensor, which utilizes electrical magnetic fields induction to sense metal objects.

(a) Proximity sensor [85]



(b) Proximity sensor internal [86]

FIGURE 3.9 Proximity sensor [85] [86].

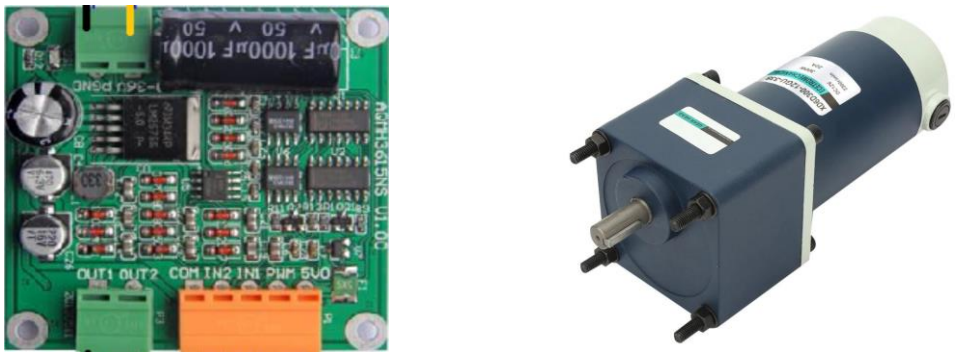
3.1.3 Actuators

To make the robot function properly, several actuators must be integrated in its structure. The robot needs to be able to move along the solar panel array and to rotate its brush for cleaning. It should be also able to lock itself on the solar panel frame in the face of severe wind. Additionally, the robot should be able to lift the brush to clear obstacles in its path (e.g., antennae, gearbox, and BHA gap).

3.1.3.1 Electrical Motor and Motor Drive

The brushed DC motor can be controlled by its input voltage, as its output rpm is propositional to its input voltage. The device that is used to control an electric DC motor is called an electrical speed controller or motor drive. It utilizes duty cycle control to change the average output voltage level and achieve electric DC motor speed control. Normally, the duty

cycle operates at frequencies higher than 1000 Hz. By being a digital controller, it has no analog output, but with such a short duty cycle (less than 1 m/sec), the switching period is shorter than the time constant of the motor, therefore, the motor will perceive this input voltage as a pseudo-analog voltage and respond to it as such. An H bridge configuration is normally used in the speed controller to control the motor for both rotation directions. This setup is used to control the path of the robot along the solar panel array as well drive the cleaning brush. Figure 3.10 (a) depicts an AQMH3615NS motor driver [87], which can take up to 15A of current at 36V of input voltage. Figure 3.10 (b) depicts a XD6D300-12GU DC motor with gearbox [88], which is rated for 300W of power at 12V.



(a) Motor driver [87] (b) DC motor with gearbox [88]

FIGURE 3.10 Motor driver and motor [87] [88].

3.1.3.2 Electrical Magnetic Locker

The TAU-S0837DL is an electrical magnetic locker [89] composed of a coil with an iron core and a spring, as depicted in Figure 3.11. The locking is done by the pushing force of the spring. When the coil is energized, it creates a magnetic field that pulls back the iron core, compresses the spring, and opens the locker. It is an industrial standard that a locker will be in the locking phase when it is not energized, and the locker used in this project does so. This ensures safety as well as reduction of energy consumption. An electrical magnetic locker can

be controlled with a general-purpose input/output pin on the main controller and a power amplifier or electrical relay.

FIGURE 3.11 Electrical magnetic locker [89].

3.1.3.3 Servo

Servos—devices with a motor, gearbox, and rotation angle feedback sensor—are widely used in automation control. They can be controlled with a pulse width modulation (PWM) signal as input, and the duration of the pulse translates to an angle value. The servo device is used to lift the cleaning brush during the cleaning process, when the brush needs to clear an obstacle. A servo can be interfaced with a general-purpose input/output pin on the main controller as well. Figure 3.12 depicts a 34109-MD servo [90] that is a combination of four components— a driving motor, a gearbox, a rotation sensor, and a control board. The servo has 190 kg/cm of torque at 12V input and has the power to lift the brush assembly when the cleaning robot encounters an obstacle.

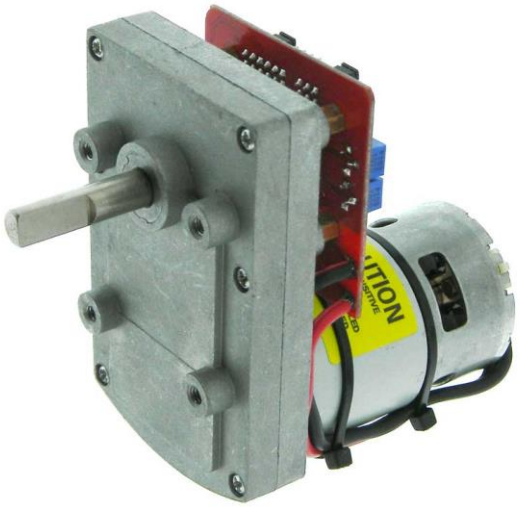


FIGURE 3.12 Servo [90].

3.2 Software

The software used in our R&D work consists of three parts. The first is the firmware development software, or IDE, for all of the WSN nodes. The second part is the firmware itself, including the MCU operating system and the applications built on top of the micro-OS. The last part is the Contiki operation system-based WSN simulator, which allows for simulation and testing of WSN performance.

This thesis uses open-source software for cost and efficiency. This software provides the tools needed to program both the WSN nodes and the robot controller.

3.2.1 Eclipse IDE

Eclipse is a powerful integrated development environment. It provides a coding interface and a toolchain to compile and build the source code into machine code, and has the capability to interface with external tools to perform programing on the target device. It also provides an interface for hardware debugging, which can be useful for firmware development. The

development process is described in Figure 3.13. The source code for Contiki is written in C language. Eclipse provides a programming interface to edit the source code and a complete tool chain to compile the source code into machine language. The tool chain contains information about the central processing unit (CPU), where the machine code will run, so that the machine code firmware and the hardware can be matched to perform the designed processes and tasks. The machine code is then transferred from the computer to the target device, in this case, to the ARM stm32 micro controller on the TIPoT development board through Jlink [63]. Jlink is hardware that is used to program a micro controller using a computer. It can be manipulated with Eclipse IDE as well. Additionally, hardware debugging can be performed with Jlink and Eclipse. This means that the code-running process on the target device can be monitored and controlled through Jlink, using a computer. Break points[4] can be set and print messages can verify the logic implemented in the micro controller using Eclipse. This makes the debugging process on a micro controller considerably easier.

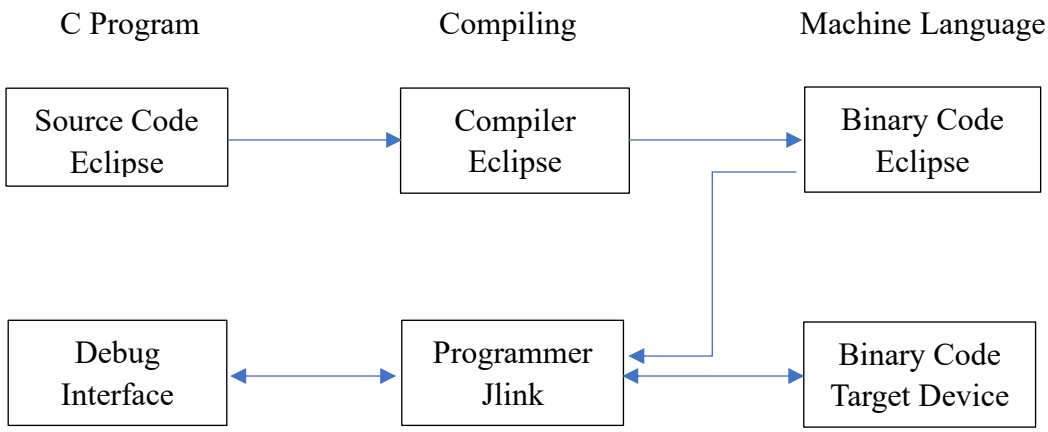


FIGURE 3.13 Firmware development process.

[4] When debugging, break points will be set up in loops or sub functions to check if this loop or sub function has been activated.

3.2.2 Contiki Micro OS

An embedded operating system is an important component for many modern applications. In fact, it exists in most of today's electronic devices. The functions of an embedded operating system, installed in an WSN node, are sensor and actuator interfacing, WSN communication, and running a user-defined process. As can be seen from this list, an embedded system will be required to deal with multiple tasks.

Contiki is an open-source MCU operation system. The benefit of running the Contiki micro-OS is that the 6LoWPAN communication protocol is already implemented within it. This makes Contiki one of the few micro-operations systems to support WSN generically.

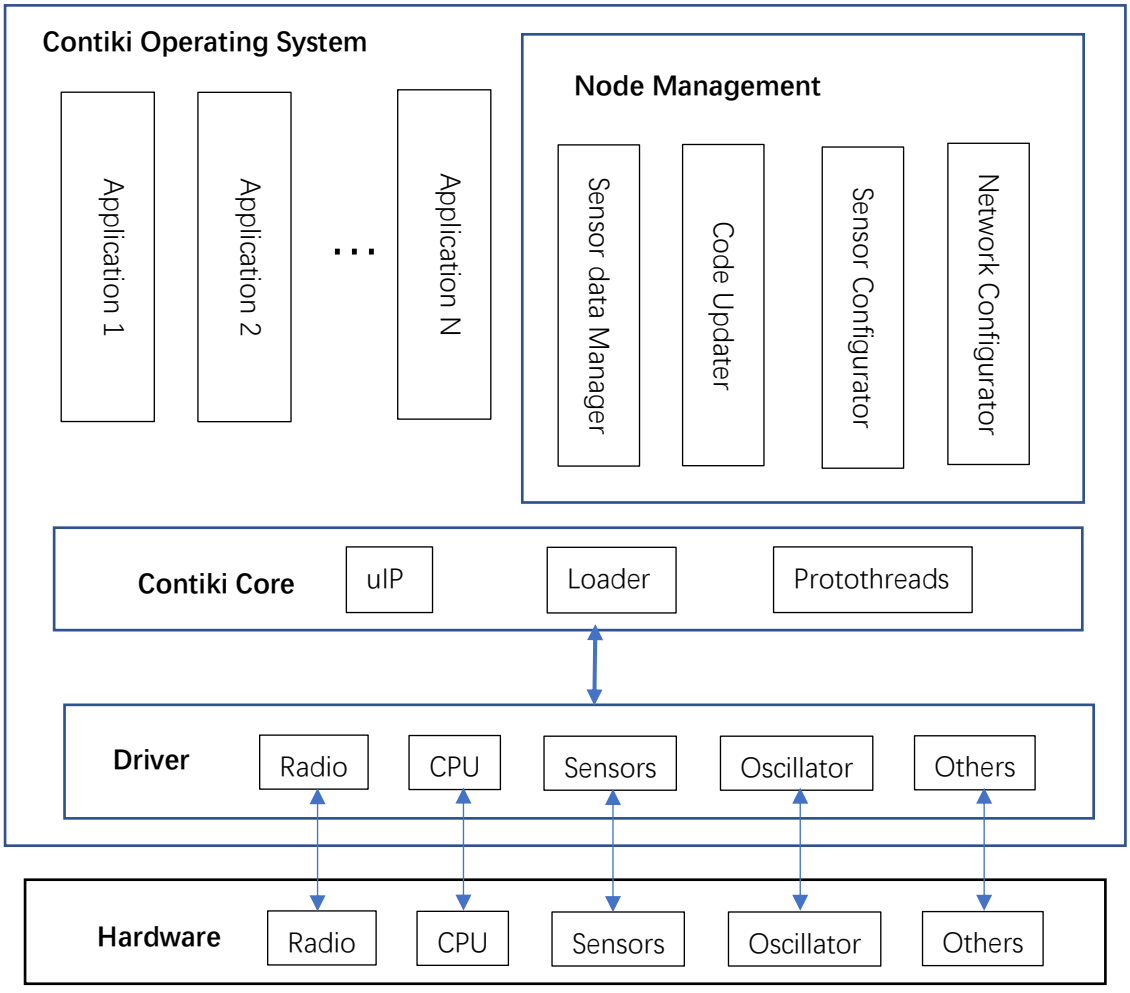


FIGURE 3.14 Contiki operating system architecture [66].

Contiki is designed for a hardware platform with limited resources—small memory, low power, narrow communication bandwidth, and low computational power. It supports the uIP TCP/IP stack [93], uIPv6 stack [94], and Rime stack [95] networking architectures [7]. Figure 3.14 depicts the system architecture of the Contiki operating system in block diagram form. We can see that its core system is built on top of the communication stack and threads, then it has drivers to operate the hardware and a management system to deal with applications.

3.2.3 The Cooja Simulator

Cooja [37] is a simulation software for a Contiki micro-OS based WSN. The focus of the Cooja simulator is to develop and test various WSN configurations and WSN applications prior to the implementation of complex and time-consuming prototypes. Cooja simulates both the hardware and software of a WSN as well as the communication environment. Cooja emulates a hardware node inside a computer, then compiles the source code into firmware and loads it into the emulated hardware to run the simulation. This means that any compiled firmware for WSN nodes, through Cooja, can be transported to a hardware node that has similar hardware specifications, requiring minimal modifications.

As indicated in Figure 3.15, the simulation interface includes four windows—the network window, the simulation control window, the mote output window, and the timeline window.

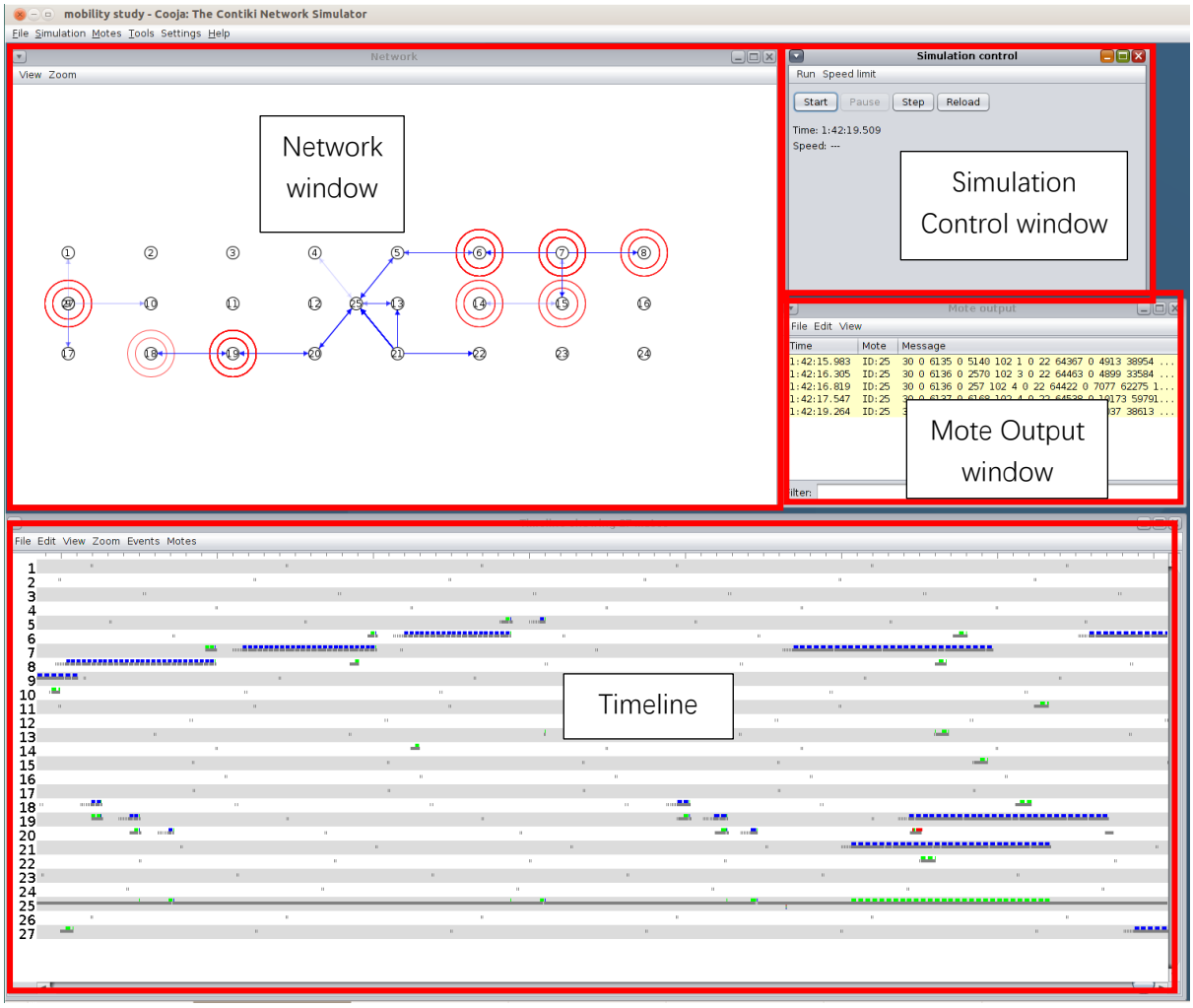


FIGURE 3.15 Cooja simulator interface.

The network window depicts and controls the physical location for each node of the network. Users can click and drag the node to a different location or set the signal strength, firmware type, and so on, by right-clicking on a node, which opens the property list. The blue arrow pointing from one node to another indicates that a packet has been sent, with the arrow pointing to the destination node. This allows the user to monitor the communication process occurring between nodes at that point in time. The simulation control window allows the user to control the simulation process, starting, pausing, stopping, and reloading a simulation. The simulation speed depends on the simulation complexity and the hardware's computation power; the user cannot accelerate the process. The simulation can be slowed down only by establishing

a speed limit, which is a useful setup when the simulation is running at extremely high speed, 20 to 30 times faster than the real clock time. Otherwise, the user cannot visualize or follow the changes that are displayed on the network window or the timeline window. The mote output window allows the user to monitor each node through debug information, such as error messages or sensor data, if it has been programed in the firmware of the simulated node (parameter monitoring is done by a collect view plugin, which is discussed in Section 3.2.3.2). The timeline window allows the user to monitor every single communication event on a timeline with a timestamp. The timestamp has a resolution of 0.1 millisecond and can capture data packets and control messages quite accurately. With the timeline window, users can monitor the order of control message exchange, understand the process of a protocol, and analyze the results.

3.2.3.1 Mobility Plugin for Cooja

Cooja has many plugins. The mobility plugin [67] is a handy tool to help better modeling WSN with moving nodes. Without the mobility plugin, the mobile sensing node must be moved manually in the network window. It utilizes a *positions.dat* file to specify each node's x and y coordinates with a timestamp. Once the mobility plugin is built and set up in the Cooja simulator, that positions.dat file can be used to manipulate the movement of a sensor node in the simulation. This allows for evaluation of the performance of the RPL protocol with mobile sensing nodes. Figure 3.6 illustrates when a mobility plugin is enabled in a Cooja simulator.

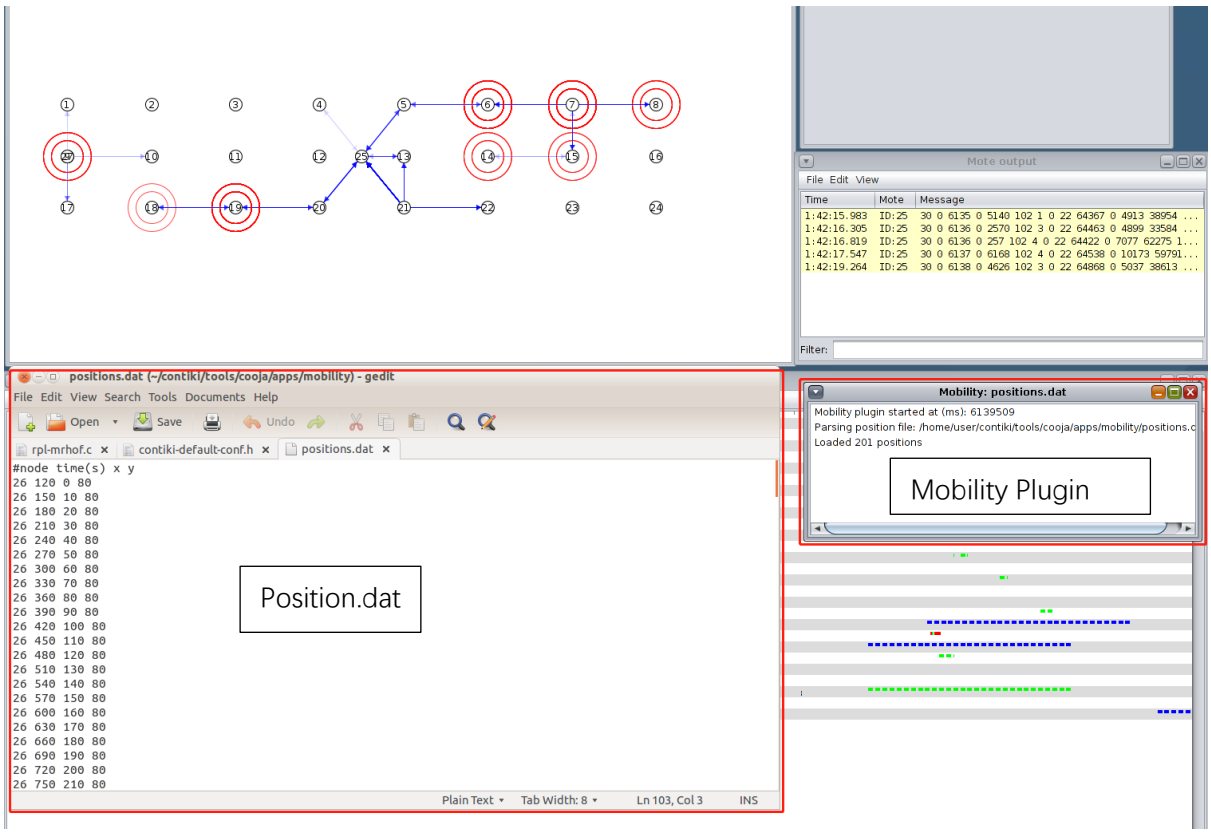


FIGURE 3.16 Cooja simulator with mobility plugin.

3.2.3.2 Collect View in Cooja for Data Analysis

Data generated during the simulation are collected and stored for analysis, and Collect view is a plugin that helps achieve this object. Using the collected data generates the network topology display, network metrics plots, and power consumption plots. The network topology display enables one to easily perceive the relationship among nodes in a WSN. The network metrics provide network-related information for a single node or several nodes, even the entire network. Several presented parameters are the number of neighbors a node has, the beacon repetition period of a node, packet loss experienced by a specific node or the entire network, and so on. The network metrics-related plots help evaluation and analysis of the performance of the RPL protocol, and of the modified versions that have been implemented. Power plots display the power consumption of each node—both instantaneous power and average power

for transmitting, listening, CPU computation and low power mode (LPM)[5] consumption are recorded; each one of them is explained in Chapter 5. Power-related plots enable assessment of the tradeoff between performance and energy consumption level.

[5] LPM is the energy consumption occurring when the node operates at low power mode.

Chapter 4

Design and Development of the Communication and Control Systems

4.1 Introduction

As described in Chapter 1, energy loss due to soiling losses and detection of solar panel defects are the two major challenges in the O&M of large-scale solar electric power generation plants. To address these challenges, a smart robotic cleaning and inspection system (SRCIS), consisting of a group of robots, is needed in the field to perform the cleaning and inspection tasks. For the SRCIS to be effective, a number of factors and constraints must be taken into account during design and development. They are the following:

1. The SRCIS solution should ensure high RoI, which requires the manufacturing, installation, and operation costs of the SRCIS to be reasonable.
2. For several reasons, including safety and security, solar electric power generation plant operators require the SRCIS to be a stand-alone system. Therefore, the robots cannot make use of the power generated by the plant; instead, they must be self-sustainable.
3. A robot should perform cleaning and inspection tasks by moving from one end of a

solar row to the other. The solar row could be very long, which makes it difficult to supply power to the robot using wires.

4. The weight of the robot must be less than 30 kg to ensure that it will not create an excessive load, resulting in bending or damage to the solar panel. The robot includes a brush, drive train, battery, and a small carry-on solar panel for self-charging, which makes weight control difficult. An energy budget must be performed to ensure the right battery and carry-on solar panel size.
5. The robots are normally scheduled to perform cleaning tasks after sunset so that the cleaning process will not impede the power-production activities of the plant. Thus, the fixed-location routing and sensing nodes must have a self-sustainable power source as well, in other words, a solar panel and a rechargeable battery.
6. Depending on the soiling rate, robots might have to perform cleaning every day. The required reliability of the cleaning and inspection robot is 95%.

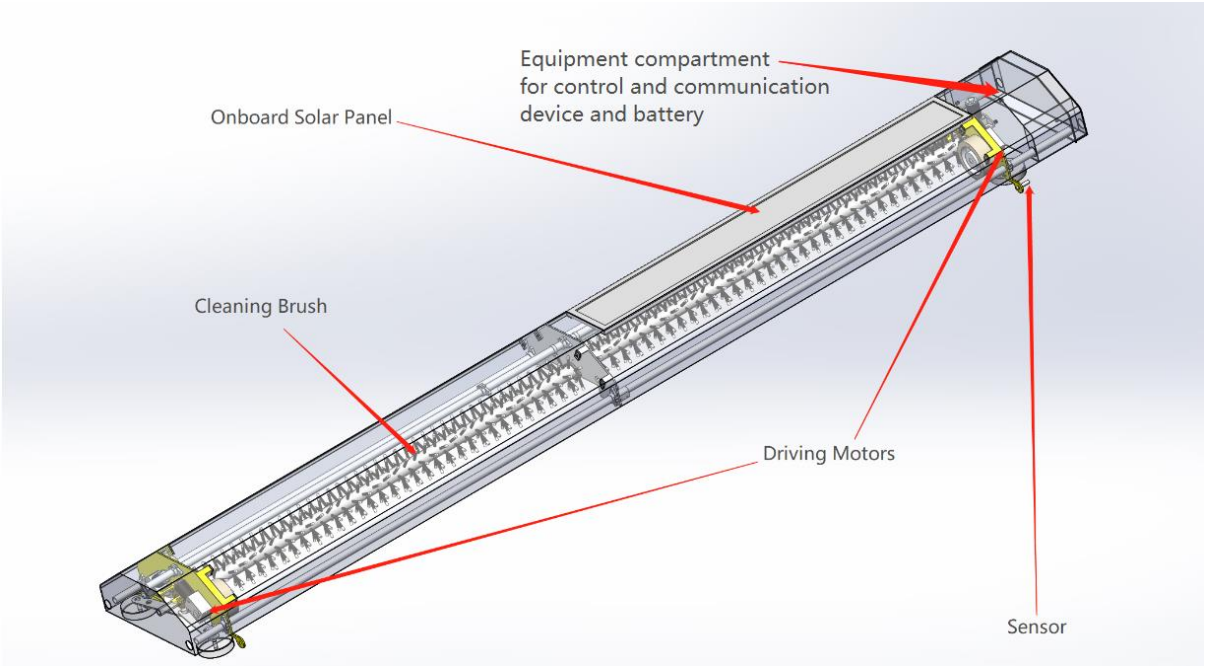
4.2 System Design

4.2.1 System Architecture

The developed SRCIS consists of four subsystems: (1) solar panel cleaning and inspection robot; (2) remote-control and monitoring system; (3) wireless network for transport of monitoring and control-related messaging; and (4) data storage/analysis and software management system.

1. Solar panel cleaning and inspection robot

The cleaning and inspection robot is equipped with a solar panel and rechargeable battery for fully autonomous operation. The robot uses a cleaning brush, which removes soiling from the surface of the solar panel by rotating around its axle. The brush is installed on a mobile body, which moves forward and backward. This mobile body is where the control and communication system is installed. As mentioned earlier in Section 3.1.2.1, in addition to cleaning, the robot performs a smart scan to detect hot spots, which significantly reduce the efficiency of the solar panels. By replacing manual inspection, it speeds up the process, reduces costs, and improves the O&M process, eliminating the time-consuming, labor-intensive manual inspection. On top of this, the ability to perform frequent inspections extends the lifespan of solar panels. Figure 4.1 (a) depicts a typical cleaning and inspection robot in a computer-assisted drawing design format. Figure 4.1 (b) depicts a functioning cleaning and inspection robot, installed on a testing rack. Figure 4.2 illustrates the cleaning brush touching the solar panel's surface and the driving wheel riding on the frame of the solar panel. They are both under the metallic cover of the robot.



(a)

(b)

FIGURE 4.1 Cleaning and inspection robot unit.

FIGURE 4.2 Inside view of the cleaning robot.

2. Remote control and monitoring system

The remote-control and monitoring system provides a user interface for the solar electric power generation plant operators to control the robot as well as monitor the data collected from the sensors deployed within the plant. Tasks include scheduling the panels' cleaning time, issuing emergency stop commands, displaying sensor data, and setting thresholds for parameters that are used to trigger alarm messages. Such messages are critical to making important decisions, such as issuing emergency stop commands to prevent damage to the robot or panels.

3. Wireless network

As mentioned earlier, communication is implemented through a WSN running a 6LoWPAN [26]. Robots are distributed over the area of the solar electric power generation plant, which could be quite broad. Figure 4.3 depicts a typical solar array row in Jiangshan, Zhejiang, China, which belongs to the China General Nuclear Power Group (CGNPG) [78]. Part of this array was used to perform a functionality test. The

solar panel rows are about 1 km in length. Fixed-location routing and sensing nodes are deployed in the field to form a multi-hop network, which provides communication capabilities across the area. More detailed information is provided in Section 4.7.

FIGURE 4.3 Typical solar array row at our test site in Jiangshan, Zhejiang, China.

4. Data storage/analysis and software management system

The sensing and inspection data collected by the robots and the fixed-location routing and sensing nodes can be transmitted back to and stored in a database on a local server or a cloud server, depending on the needs, requirements, location, and tele-connectivity infrastructure accessible by the solar electric power generation plant. The maintenance or management person has the flexibility to use a computer, tablet, or smartphone to control and monitor all robots in the system through a Local Area Network (LAN) to efficiently perform O&M tasks. Management staff can also access and analyze historical data to make informed decisions while assisted by an expert system.

The system architecture is depicted in Figure 4.4. We see that the system can be devised into two type of networks; on the right side is the WSN for sensing and robot control and on the left side is the TCP/IP network for the site operators to access the data.

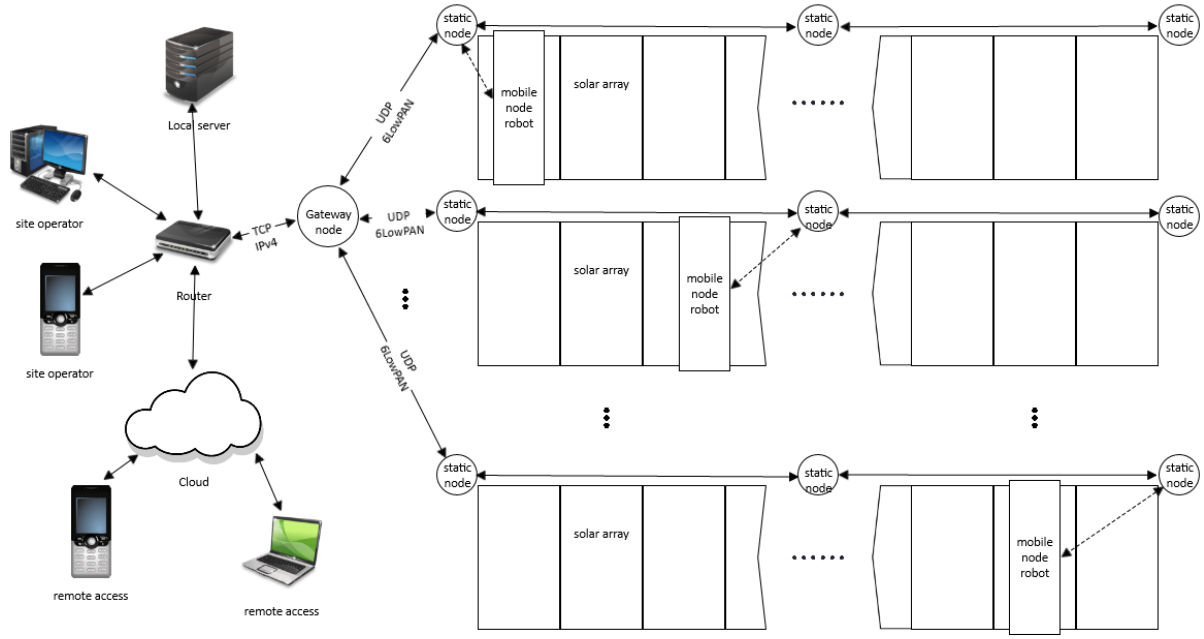


FIGURE 4.4 SRCIS communication system architecture.

4.2.2 Robot Control System

Due to the constraints of weight, water resistance, and space available for the installation of electronic components, the control and communication module of the robot is implemented on a single PCB and is placed in an IP65 rating [68] box together with a maximum power point tracking system (MPPT) [69], voltage regulators, and a motor driver. The MPPT is a charging system that charges the cleaning robot's battery through the onboard solar panel and can adjust its output current and voltage to achieve the maximum power output to charge the battery. The box provides an aircraft connector to connect with the onboard solar panel, motors, and sensors. Figure 4.5 illustrates the design of the robot control system. The onboard solar panel, MPPT unit, battery, and voltage regulators are included in the power and power management block. The control system consists of a control and communication module, power and power management module, motor driver, motors, and sensors.

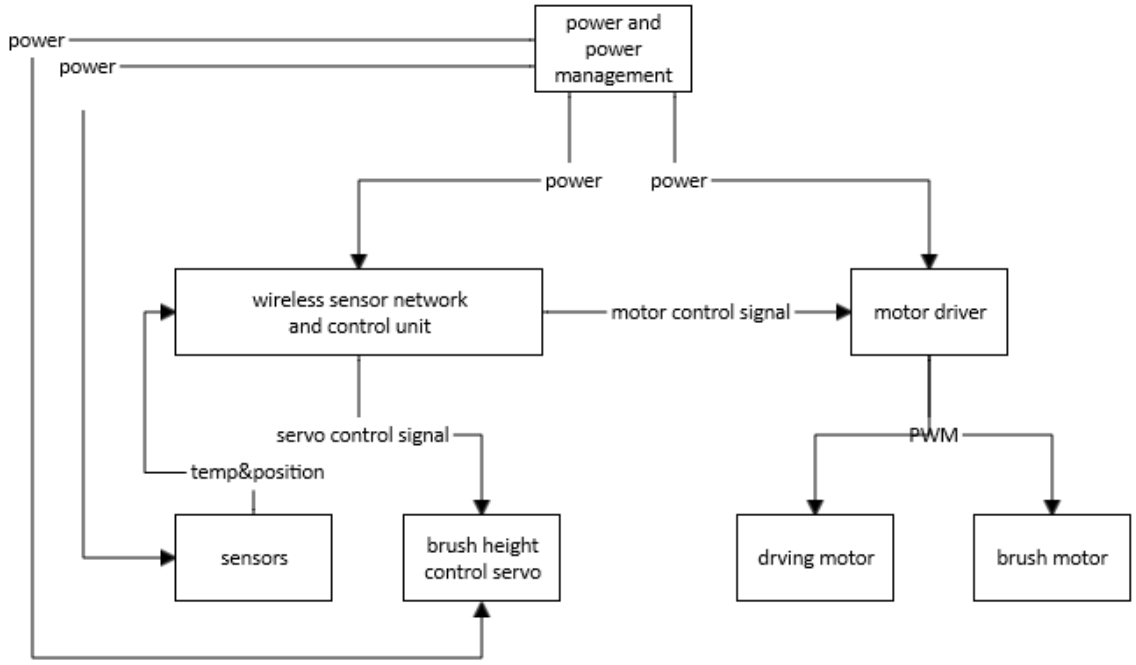


FIGURE 4.5 Block diagram of the robot control system.

1. *Wireless sensor network and control module*: A TIPoT mote board [62] is programmed to be the control and WSN module. The TIPoT mote board has an ARM Cortex-based STM32F103 MCU [64] with wireless communications over an IEEE 802.15.4 compliant ATMEL AT86RF231 transceiver [65]. The STM32F103 MCU is powerful enough to handle the required sensing and control tasks in addition to the interactions with the ATMEL AT86RF231 transceiver for WSN communications. The choice of a development board with both MCU and RF transceiver simplifies the hardware design while satisfying the system requirements for control, sensing, and WSN communications.
2. *Motor driver and motors*: The robot moves along the solar panel array to perform the cleaning and inspection task. Special wheels are used on the robot to enable it move smoothly on the frame of the solar panels, as indicated in Figure 4.6. The driving system

includes the motor driver and the driving motor. The motor driver is also responsible for providing regulated power to the brush motor to rotate the brush for its cleaning task. A servo motor is used to raise and lower the brush when the robot encounters obstacles along the moving path.

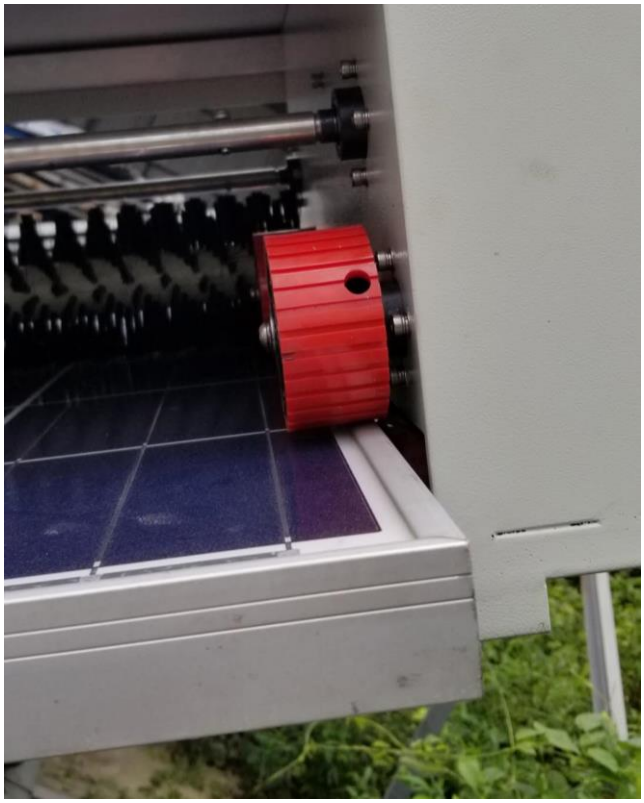


FIGURE 4.6 Robot wheel riding on the solar panel frame.

3. *Sensors*: A set of sensors are installed in the robot for a variety of purposes, including voltage and current sensors for monitoring the energy level of the battery, proximity sensors for detecting obstacles and automating the returning and parking of the robot, and temperature sensors for detecting hot spots. The sensors are installed in appropriate locations on the robot to collect relevant data and send to the MCU for processing. Some of the data are sent via the WSN to the database for analysis and storage.
4. *Power and power management module*: This module supplies power to all components

of the robot, which include a solar panel, an MPPT, a rechargeable battery, and voltage regulators. The solar panel charges the battery through the MPPT when the robot is parked on the docking station during the daytime. The battery supplies power to all other components. The voltage regulator produces the desired voltage required for the operation of each component.

4.3 Design and Development of Power Management Module

Although solar electric power generation plants generate electricity, the SRCIS cannot use the power generated from the solar panel arrays to power itself for safety and security reasons. The SRCIS must be an independent, stand-alone system; this robot relies on its own onboard solar panel to generate power for itself. Therefore, the design of a power supply module is extremely important in ensuring the robot's self-sustainable operation. A number of factors must be taken into account. On one hand, the power should be sufficient for the robot to perform round-trip cleaning on the solar row. On the other hand, the cost, size, and weight of the onboard solar panel and rechargeable battery are limited. Figure 4.7 depicts our design of the power supply module.

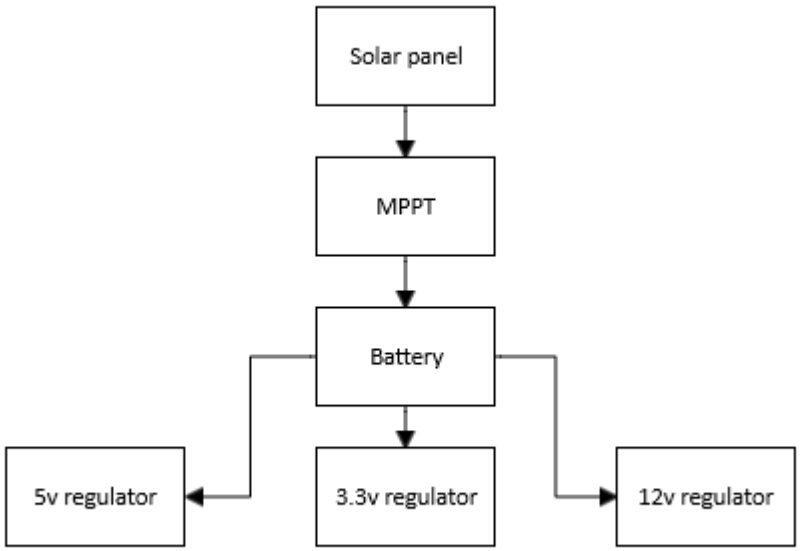


FIGURE 4.7 Block diagram of the designed power supply module.

During the daytime, the robot's solar panel generates power and charges the rechargeable battery through the MPPT module. Then, the battery supplies the electric energy for the robot to perform its tasks after sunset. The power supply module of fixed-location routing and sensing nodes has a similar design, but the calculation of the power budget is different and is based on their own functionality and energy consumptions. In the following sections, the calculations of the power budget for both robots and fixed-location routing and sensing nodes are presented.

4.3.1 Power Management Module of the Mobile Sensing Node and Cleaning Robot

The mobile sensing node is installed on the cleaning and inspection robot and is used to control the robot and collect sensor data. There are strict constraints on the size and weight of the robot, with the onboard solar panel and battery accounting for a significant portion of the total weight. To ensure that the power supply module is efficient and effective, the following design factors should be taken into consideration.

1. The rechargeable battery installed in the robot should be able to supply sufficient power for the robot to perform daily round-trip cleaning on a solar row up to 1 km in length (a total travel distance of 2 km).
2. The onboard solar panel should be able to generate sufficient power to fully charge the battery under normal weather conditions.
3. Since the power capacity of a solar panel and battery is proportional to its size and weight, the onboard solar panel and battery should not be oversized, which could lead to an overweight robot and increased costs.

To achieve these design goals, an energy budget analysis must be undertaken to calculate how much energy the robot needs to perform its cleaning and inspection tasks and to determine the size of the battery and the onboard solar panel.

The highest energy consumption occurs when the robot is moving and/or the brush is cleaning. We need to collect information for motor power consumption. Since the rated power indicates only the capability of the motor, while the actual energy consumption of the motor depends on the load and the robot's working condition, we must measure the actual energy consumption that occurs when the robot is performing its tasks. For example, if a motor is rated 300W at 12V, that does not mean the motor will consume 300W whenever it is connected to power. We can calculate that the full load current for this motor is 25A. It could consume as little as 0.1A at no load and as much as 25A at full load. The steady-state current draw was measured for calculating the continuous power output of the motor. This is the power output that the motor will deliver for most of the time when the robot is working. It is noted that the motors may consume more power during start-up and reversing, but these are all short-lasting

events, insignificant compared to the steady-state power consumption. A 12V (V_R) system is used, and the average running current for driving the motors and servo is measured to be 9.4A (I_{RA}), resulting in 112.8W ($P_{RA} = V_R * I_{RA}$) instantaneous power draw. The robot is moving at a speed of 20 meters per minute (v_R) and the round-trip cleaning distance in total is 2 km (L_T). It takes about 100 minutes ($T_T = L_T / v_R$) for the robot to complete a cleaning cycle, which consumes 188WH ($W_R = P_{RA} * T_T$) of energy.

In addition to the energy consumed by the motor, the control and communication modules are also consuming energy. The current is measured at approximately 0.3A (I_M) under 5V (V_M) working voltage, which leads to an additional 36WH ($W_M = V_M * I_M * 24hr$) of power consumption. The four position sensors on the robot are working at 12V (V_{PS}) and consume 40mA (I_{PS}) current, which creates 3.2WH ($W_{PS} = V_{PS} * I_{PS} * T_T$) of energy consumption. The temperature sensor consumes the least energy, approximately 0.2WH (W_{TS}). Therefore, the total energy consumption for the robot to complete one cleaning cycle is approximately 227.4WH ($W_{TM} = W_R + W_M + W_{PS} + W_{TS}$). Adding a small margin, the designed power supply module should be able to supply a minimum of 240WH (W_{BM}) of energy.

The onboard solar panel generates power for the system, which has a size limitation due to the robot's size and weight constraints. Normally, the power rating of a solar panel is proportional to its size, and the solar panel accounts for a significant portion of the total weight. As a general guideline, the total solar energy collected in a day is equivalent to power production under about four hours (T_C) of full sunlight [70]. Therefore, a 60W (P_S) solar panel is used for the robot, which has a daily power generation of approximately 240WH ($W_S = P_S * T_C$).

The weather is harsh at most solar electric power generation plants. For example, some plants in the Middle East require the robot to work under a temperature range from minus-10C to 70C. A lithium ferro phosphate (LiFe) battery [71] is selected for use onboard the robot. This battery has a high energy density, which means for the same size and weight limit, LiFe will contain more energy than a lead acid battery [72]. Additionally, a LiFe battery is robust; it can be overcharged or over discharged without significantly reducing the battery's life cycle. Moreover, the latest generation of LiFe batteries can have an operation range from minus-60 to 60C. As a general guideline, a LiFe can discharge only 65% of its rated capacity to maintain its operational voltage. Therefore, a 350WH ($C_{BM} = W_{BM} / 0.65$) battery is needed to provide the 240WH energy required for the robot's daily tasks, without falling below the 65% discharge point. Since the system is operating at 12V, a 4S LiFe battery is ideal for the system, as its operational voltage will be 14V ($V_B = V_{BS} * 4$), which is achieved by placing four 3.5V (V_{BS}) batteries in a series. Therefore, a 4S LiFe battery with 25AH (amp hours) capacity is used for this robot system; this is simply obtained by dividing 350WH by the rated voltage, 14V. The extra capacity of the battery can serve as a buffer for balancing the power generation and consumption. It is noted that simply increasing the capacity of the battery does not translate to noticeable additional daily operational time for the robot, since the limitation comes from the volume of energy that the robot's solar power source can collect within a 24-hour period. Moreover, increasing the size of the battery will increase the weight of the robot, which is counterproductive—the weight is more compared to whatever limited additional time it can achieve.

4.3.2 Power Management Module of the Fixed-Location Routing and Sensing Nodes

Fixed-location routing and sensing nodes are fixed-position nodes that form a grid in the solar electric power generation plant, collecting sensor data and acting as access points for the mobile sensing nodes.

The current drawn by a fixed-location routing and sensing node is approximately 0.3A (I_{FM}). Therefore, the communication and sensor data collection module of the node consumes about 1.5W (P_{FM}). The attached sensors have a rating of 3.3V/30mA, which accounts for a very small portion of the energy consumption. The total operation power consumption for the fixed-location routing and sensing node is less than 2W (P_F). Consequently, 48WH ($W_{BF} = P_F * 24hr$) of energy is needed for daily nonstop operation. The solar panel will receive a four-hour-equivalent charge daily at its power rating, thus, a 15W 12V solar panel is chosen for the fixed-location routing and sensing node. A 2S LiFe battery is used, which provides a working voltage of 7V ($V_{BF} = V_{BS} * 2$). A step-down voltage regulator (5V) is used to provide the correct voltage to the fixed-location routing and sensing node. The battery must provide the node with energy sufficient to keep the node operating for 12 hours (during the night) without charging in the meantime. This, in turn, keeps the node functional on a 24-hour basis (day and night). Additionally, considering the need for some extra capacity to compensate for cloudy or rainy days (during which the sunlight intensity is reduced, along with the volume of produced energy) and 35% residual energy for maintaining its operational voltage, we have calculated that a 92WH ($C_{BF} = W_{BF} / 0.65$) battery is needed. A 14AH 2S LiFe battery [71] is chosen for the fixed-location routing and sensing node.

4.4 Smart Control Module

The smart control module processes sensing data and received commands, and enables the cleaning robot to function autonomously. The developed functions include:

- Locking and unlocking the robot from the docking station;
- Forward cleaning;
- Reverse cleaning;
- Gap and obstacle detection;
- Brush raising; and
- Hot-spot detection.

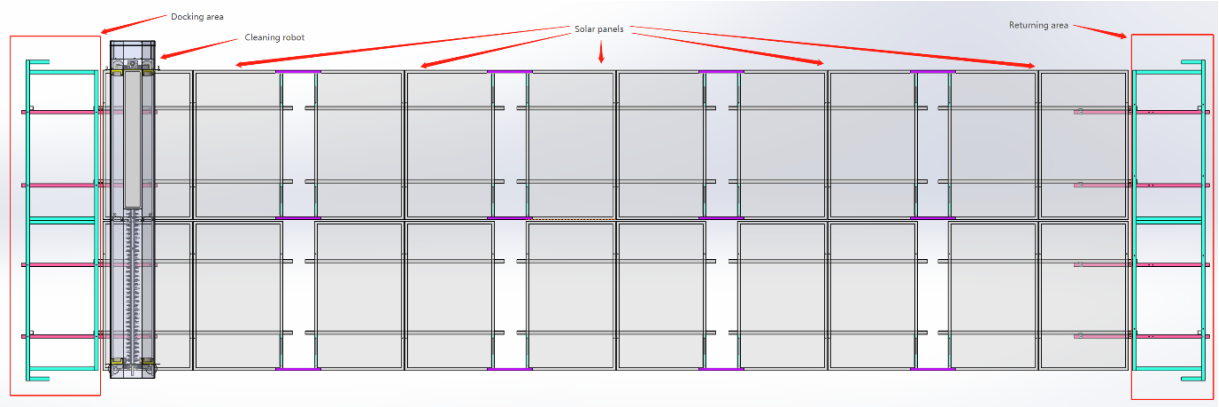


FIGURE 4.8 A shortened solar array row with docking and return section and cleaning robot.

Figure 4.8 illustrates a shortened solar array row with a docking and return station on each end of the row. The robot parks at the docking station while not performing any tasks. The return station is attached to the other end of the solar array to protect the robot from falling off the solar array and provides space for the robot to perform cleaning and inspection tasks at the end of the solar array row. When the robot reaches the end of the return station, it will stop and then begin the cleaning or inspection process in the reverse direction, toward the docking

station.

Figure 4.9 and Figure 4.10 illustrate the two major working processes of the robot, namely, the robotic cleaning process and the hot-spot detection process. The robot cannot perform these two tasks at the same time. The cleaning process usually takes place after sunset, while the hot-spot detection process is performed when the solar panels are generating power during the day. Figure 4.9 illustrates the cleaning process, which is performed daily or according to the needs of the solar electric power generation plant operator. Upon receiving the "start" message, the robot will determine whether it is in the docking area. If the robot is in the docking area, it will begin the cleaning process right away; if not, it will go back to the docking area, then start the cleaning cycle. The robot moves along the solar row from one end to the other, its roller brush rotating while the robot is moving, brushing dust off the solar panels. There are cases when the solar row can have multiple of obstacles, such as a gearbox and/or bearing housing assembly. When the robot's sensors detect an obstacle on the solar row, the control module will receive a signal from the sensors. The robot then stops and raises the brush to pass over the obstacle. After the robot crosses the obstacle, it lowers the brush and continues the cleaning process. When the robot reaches the end of the solar row, metal proximity-based position sensors will be triggered. The robot stops and then begins moving in the reverse direction toward the docking station. A similar cleaning process is performed until the robot returns to the docking station and locks itself to the docking station's frame. During the entire cleaning process, if the robot receives an emergency stop (Estop) command, it will perform an emergency "shutdown," in other words, stopping and locking itself on the frame.

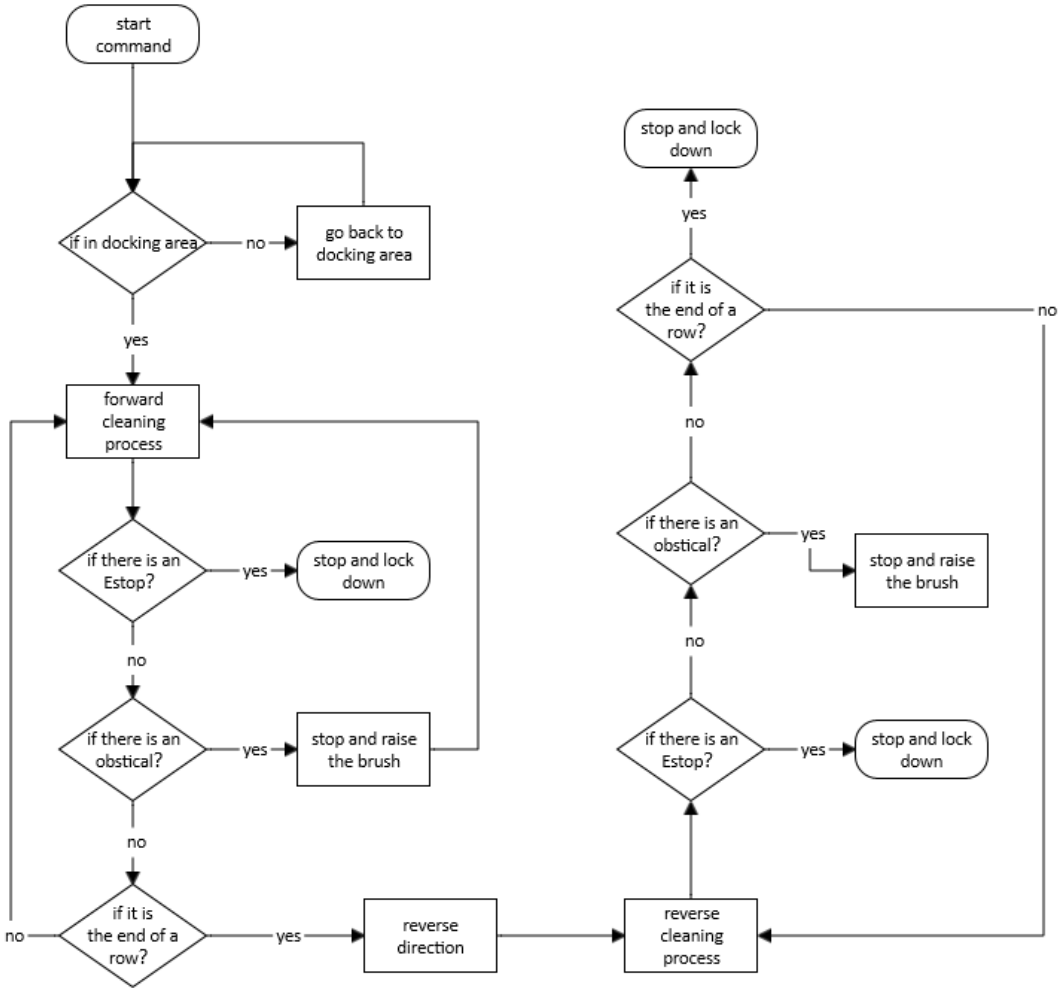


FIGURE 4.9 Robotic cleaning process.

The hot-spot detection process is illustrated in Figure 4.10. This process is performed once a month for regular maintenance or as needed. Upon receiving a start message, the robot performs the starting-point check as in the cleaning cycle, then it raises its brush with a servo motor system and moves along the solar row to scan the solar panel with a non-contacting temperature sensor that measures the temperature of the solar panel with an infrared thermometer. The use of the non-contacting sensor allows the robot to detect abnormal temperatures while moving, thereby avoiding scratching the surface of the solar panels. The collected temperature-sensing data are transmitted to the server via wireless network. During

the automation tasks, the robot may receive an emergency stop command issued by site operators if they receive warning regarding potentially hazardous conditions, such as structure failure, fire, or strong wind. Emergency-stop functionality is also available for the hot-spot detection process.

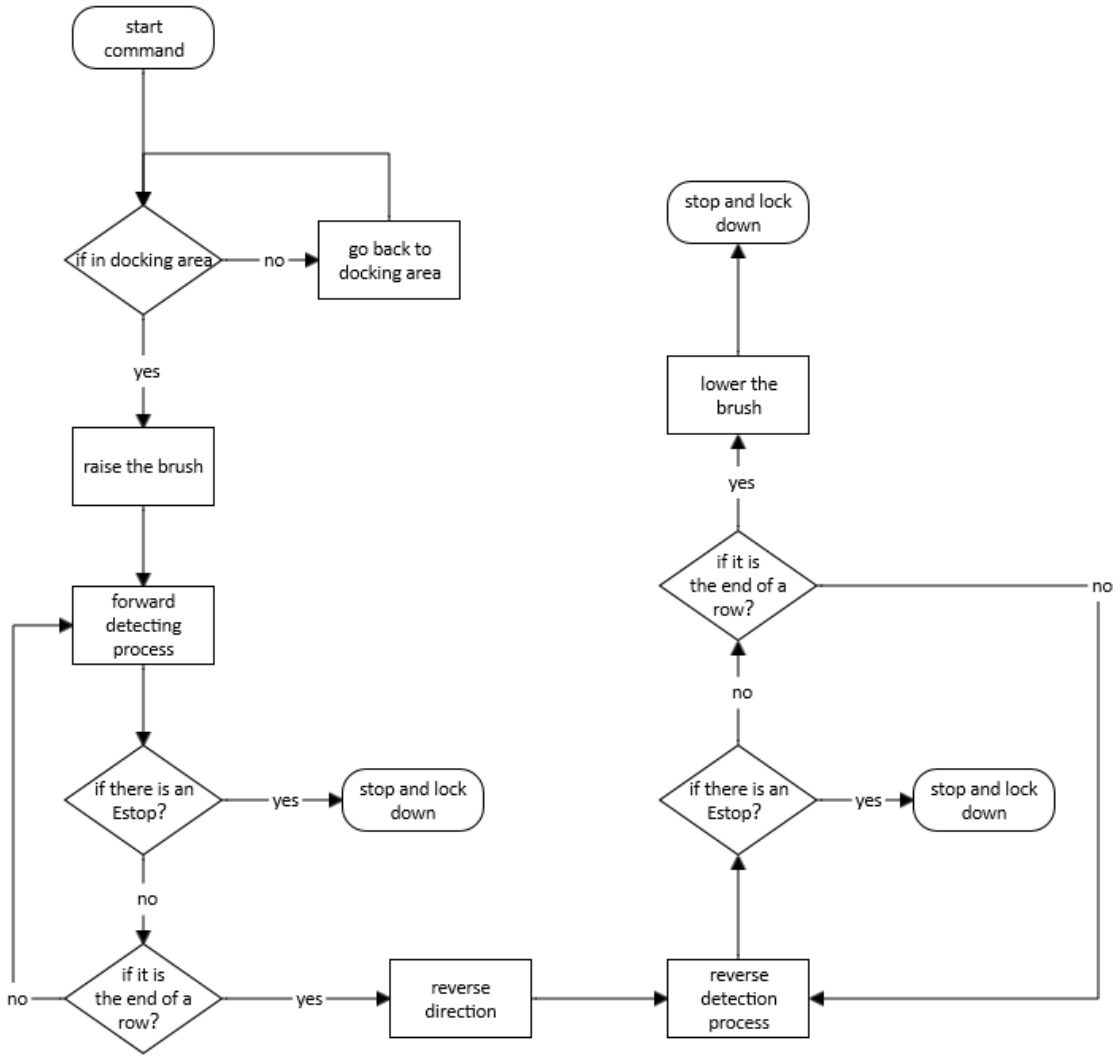
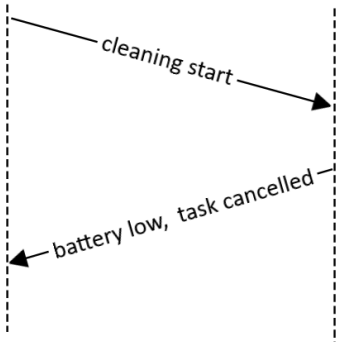
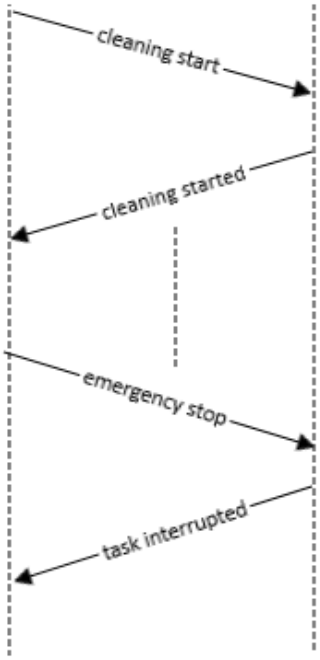


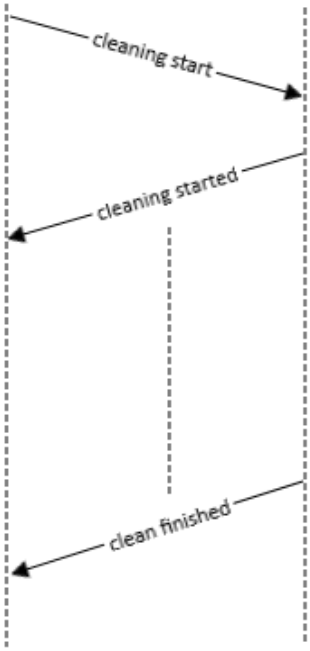
FIGURE 4.10 Process of hot-spot detection.



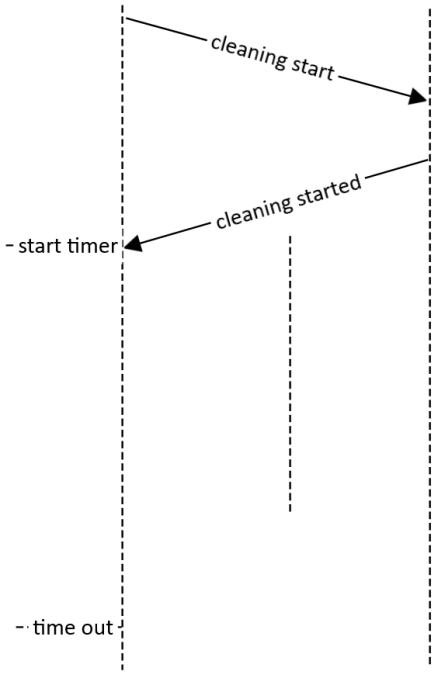
(a)



(b)



(c)



(d)

FIGURE 4.11 Communications between control center and robot.

Other than the cleaning and hot-spot detection processes, the smart control module is also responsible for checking and reporting the status of the robot. When the robot receives a start command through the WSN, it begins the cleaning or inspection process. The robot first performs a battery check. If there is insufficient energy in the battery for a working cycle, it will not begin the cleaning or inspection process; instead, it will issue a message to the local server, stating that the process will not start due to low battery (see Figure 4.11 (a)). If the battery check is passed, the robot will begin performing the process. During the process, if any emergency-stop command is received, the robot will terminate the cleaning or inspection process, lock itself down on the solar array frame, and send a message to the server stating that the process was interrupted. The robot will then wait for a command (to be issued by the operator) or for manual intervention, as indicated in Figure 4.11 (b). If the process completes without incident, the robot will send a process-completed message to the server after it returns to the docking station, as illustrated in Figure 4.11 (c). To deal with a situation in which the robot breaks down during the cleaning or inspection process, a timeout mechanism has been implemented. A normal cleaning or inspection process should finish within a certain period, depending on the average speed of the robot and the length of the solar row. Upon receiving the process-started message from the robot, the server will begin a timer to monitor the cleaning time. The process-completed message will reset this timer. After the timeout is triggered, which means the robot failed to complete the process in time, the server will mark this process as not finished due to unknown issue, as indicated in Figure 4.11 (d) and will send an alert to the plant operator.

4.5 Design and Development of Communication Network

As described in Section 4.2.1, the communication between the robots and the server is achieved by interconnecting various networks of heterogeneous technologies, for example, WSN, LAN, and/or Wide Area Network (WAN), depending on what is available in the locality of the plant.

4.5.1 Wireless Sensor Network

Large-scale solar electric power generation plants have solar rows spread over a large geographic area. Robots are distributed in the wide area of the plant. As discussed in Section 4.3, the system needs to be self-sustainable and must comply with the weight limit. Therefore, in this scenario, a low-cost, low-power, low-rate WSN is a useful choice. Wireless routing nodes can be deployed implementing mess networking and packet forwarding to provide wireless communication coverage over the area.

Other options could be WiFi and cellular networks. WiFi is designed as LAN technology and can support high data rates. However, its network interfaces have significantly higher energy consumption compared to those designed for sensor nodes, for example, ZigBee. Additionally, it is difficult to interface with a micro controller operating system. A cellular network (if available) is a good option for connecting the gateway node to the internet, but if a connection between every single node and the cellular network is desired, the cost in terms of hardware and service fees is remarkably high.

Solar power plants are located in many different countries, and, in the future, their

proliferation will increase further. Therefore, the wireless networks used should be based on widely accepted standards and, if possible, should use unlicensed frequency bands. The WSN in the SRCIS is based on the IEEE 802.15.4 standard, [56] and operates at a 2.4GHz frequency band.

The SRCIS should have a seamless internet connection. Connectivity within the plant is provided by the WSN, and the WSN moves traffic between the SCRIS and gateway. A 6LoWPAN is designed to meet the communication needs of WSN nodes, while taking into consideration the limitations associated with such devices. It also includes beneficial IPv6 features, such as the ability to use a highly scalable addressing scheme, stronger security, and full internet compliance. In the network layer, RPL is used as the routing protocol. Three key features of this service must be addressed by the deployed WSN.

1. *Data rate*. The data rate required for sending sensing data and receiving control commands is light. Sensor data can be as low as 10 bits, and the sensor update frequency can be several minutes; thus, the 80-byte payload size of a packet is sufficient for transporting sensing data and/or control commands [23]. Figure 4.12 indicates a 6LoWPAN packet structure and is what is being used in this thesis.

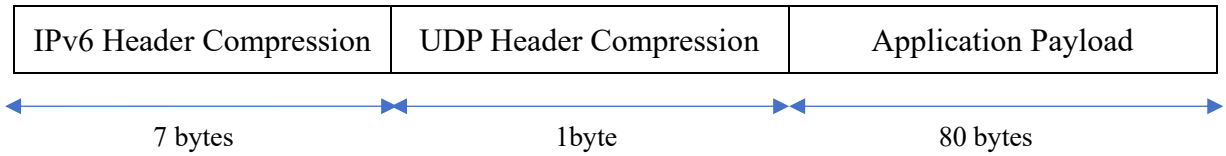


FIGURE 4.12 A simple 6LoWPAN packet structure. [23]

2. *Scalability*. Some solar energy generation plants cover large areas; thus, the network nodes must be connected with each other and form a chain, tree, or mesh topology to perform multi-hop communication, allowing data to be passed along the routing path

and reach the gateway node. RPL [19] is the default routing protocol for a 6LoWPAN, and it is designed for IoT applications, with low energy consumption and low computation power in mind. It can be implemented in a micro controller-based system. It can quickly initialize the DODAG tree formation process, and allows nodes to enter and leave the network without considerably changing the remainder of the network topology. Changing the network topology means re-calculating the routing path, which consumes computation power and energy. An RPL also contains two methods to handle network failures: global and local repair [19]. This allows the network to maintain its functionality and look for optimized routing solution at the same time.

3. *Energy consumption and range*. These two factors are considered together because it is not meaningful to just talk about transmission range without considering the energy consumption. The transmit power can be increased to gain wider single-hop coverage at the expense of faster dissipation of energy. The standardized link layer and physical layer for WSN is IEEE 802.15.4, with 2.4GHz carrier frequency and mW level power consumption. The TIPoT board can achieve a line-of-sight communication range of approximately 200 m with transmit power level of 1 mW [62].

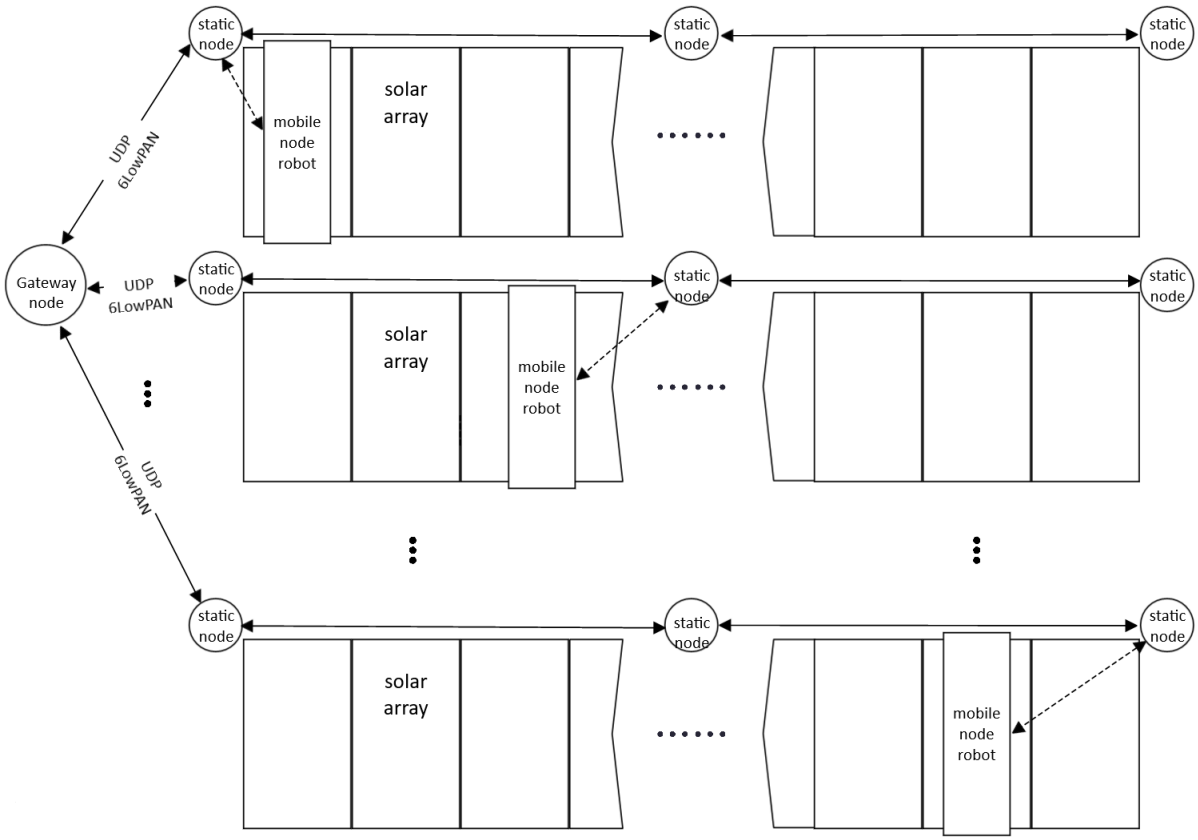


FIGURE 4.13 Gateway, fixed-location routing and sensing node and mobile sensing node arrangement diagram.

The WSN part of our SRCIS consists of one gateway, a number of fixed-location routing and sensing nodes, and robots as mobile sensing nodes, as depicted in Figure 4.13. The location of the gateway node must be planned carefully to achieve positive performance for the entire network. This is because an appropriately chosen location of the gateway can generate shortest-path links from the gateway to farthest-away leaf nodes that have relatively short length, as well as produce a balanced distribution of nodes in terms of the number of children nodes per parent node. The topology change affects the performance of the network, as demonstrated by the performance evaluation results, which are presented in the next chapter. Considering a row of the solar panel array as an example, a row of solar panel array could be as long as 1 km, and the robot must perform a round-trip cleaning to ensure a satisfactory cleaning result. This means that the robot must travel 2 km in total for one cleaning cycle. During this process, the

robot must remain connected to transmit sensor data, mainly the hot-spot detection of the solar panel array, and take control commands such as emergency stop. Fixed-location routing and sensing nodes are placed along the solar panel array to connect with the mobile sensing nodes and route the sensor data or control commands. These fixed-location routing and sensing nodes form a DODAG [19] type of routing topology, through which they route data packets to the gateway as well as transport commands to the nodes. Mobile sensing nodes attach themselves as leaf nodes to the DODAG routing tree, thus connecting themselves to the gateway.

4.5.2 Design and Development of Network Nodes

There are three types of nodes in the WSN of the SRCIS, namely, gateway node, fixed-location routing and sensing node, and mobile sensing node. They have different hardware and software designs that better meet their specific functionalities, but they also have similarities, such as the embedded operating system, RF, and power supply. Figure 4.14 indicates their similarities and differences.

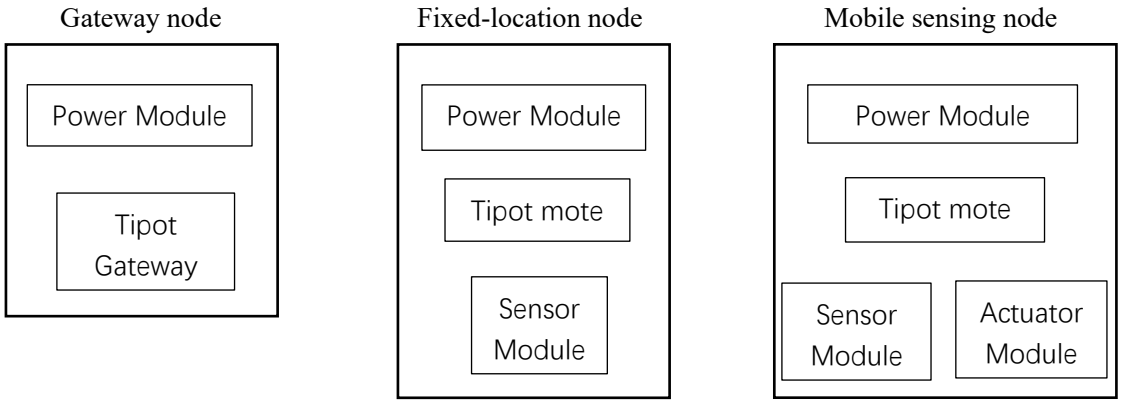


FIGURE 4.14 Node's similarities and differences.

1. Gateway Node

The task for the gateway node is to bridge two types of networks together; in this case,

the ethernet and the wireless sensor network (see Figure 4.15).

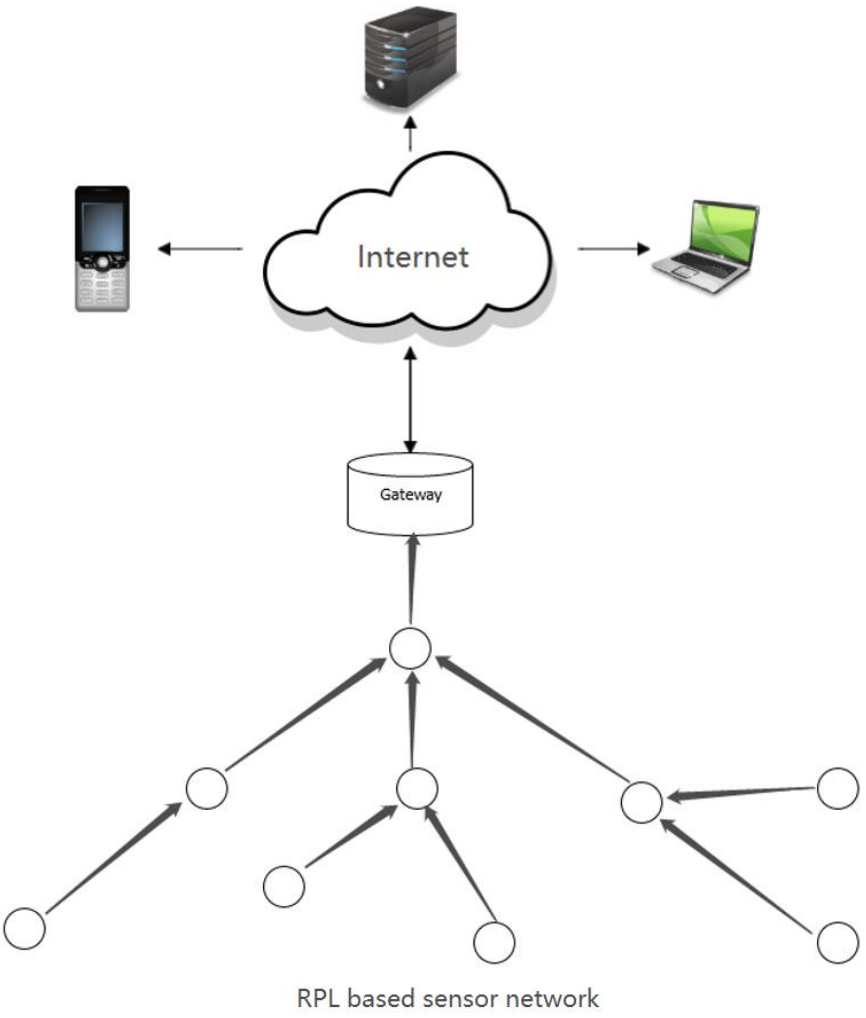


FIGURE 4.15 Gateway node bridging the wireless sensor network and internet.

In Section 2.4, we reviewed the RPL protocol and the concept of DODAG. Here, we move to the bridging between the RPL-based WSN and ethernet. The data-flow diagram is illustrated in Figure 4.16 [73]. The RF client is an application running in the Contiki OS of a WSN node. The RF server and gateway are two applications running in the OS of the WSN gateway node or, as it is also known, the border router. The gateway is located within the red box. The gateway node is responsible for translating 6LoWPAN packets to IPv4 packets following NAT64 [27] and distributes routes for nodes to interact with the ethernet [74]. From Figure 4.16, we can

see that, inside the WSN, user datagram protocol (UDP) is used for the transport layer of the network protocol stack. The connection between gateway and the server uses the transport layer protocol (TCP).

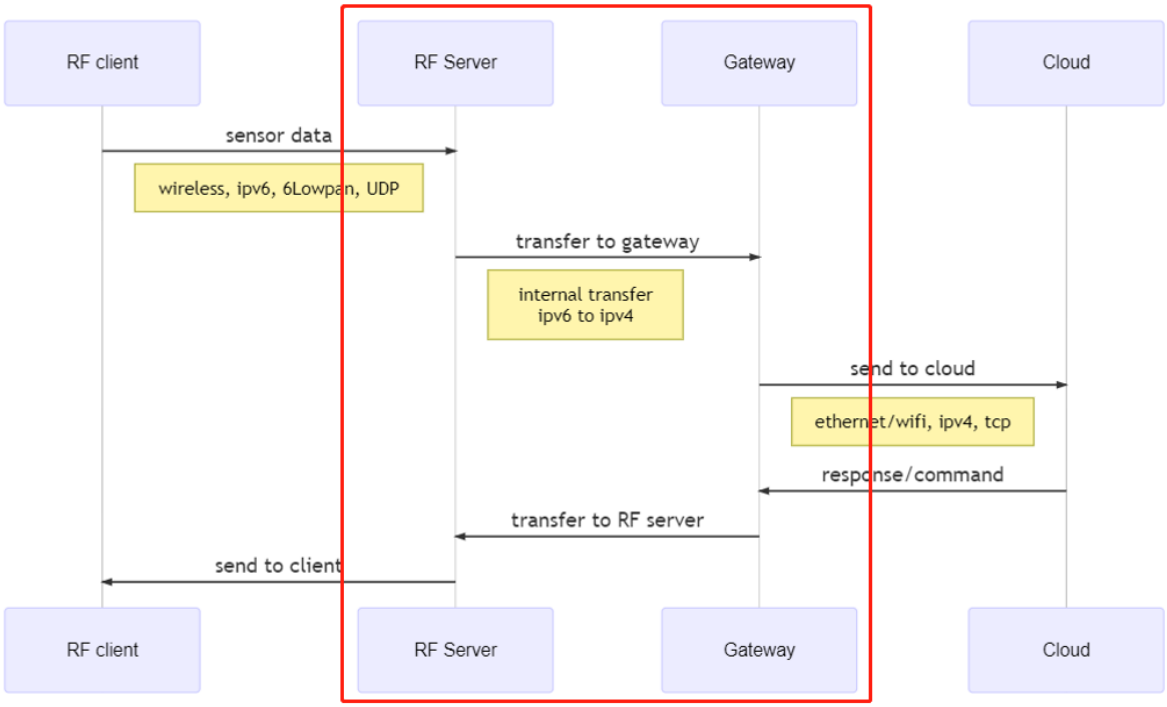


FIGURE 4.16 Border router platform diagram.

2. Fixed-location routing and sensing nodes

The hardware of the fixed-location routing and sensing node includes the sensor, the TIPoT WSN board, and a power supply module consisting of a 15W/12V solar panel, an MPPT, a rechargeable battery, and several voltage regulators. Fixed-location routing and sensing nodes are placed in the solar electric power generation plant at fixed locations to form a grid. The locations of these nodes are planned carefully to establish a reliable network. The maximum line-of-sight communication distance for a node is approximately 180 meters. This is determined by an experimental test in the solar electric power generation plant, in which the field functionality test was conducted. Considering that there may be interference, object blocking, fading, and some margin, a 100- to 130-meter separation was used in our deployment.

The sensors for the fixed-location routing and sensing nodes include a temperature sensor and a structure load sensor. Some fixed-location routing and sensing nodes also have sensors for wind speed and direction, as these data should be monitored constantly to ensure a safe wind condition for the robots to perform cleaning of the solar panel rows.

The tasks for the fixed-location routing and sensing nodes are to form a WSN grid, which can route the data packets from all nodes (with mobile nodes included) to the gateway and vice versa. The sensors attached to the fixed-location routing and sensing nodes generally do not have strict requirement for real-time communications. This allows the collection rate of sensor data to be lowered. Normally, one packet per minute is considered excessive for atmosphere temperature monitoring; even a five-minute interval between packets is acceptable. Structure tension or load data can be collected at even longer intervals.

Five sensors are attached to the fixed-location routing and sensing node for this project, and more sensors can be added if other applications require so. Each fixed-location routing and sensing node is programmed to collect data from each attached sensor in a round-robin manner, moving from one sensor to the other every one minute. The process is described in Figure 4.17. This strategy allows each sensor to update its information every five minutes. The aggregate sensor data packet generation rate that each fixed-location node generates is one sensor data packet per minute, as indicated in Figure 4.17.

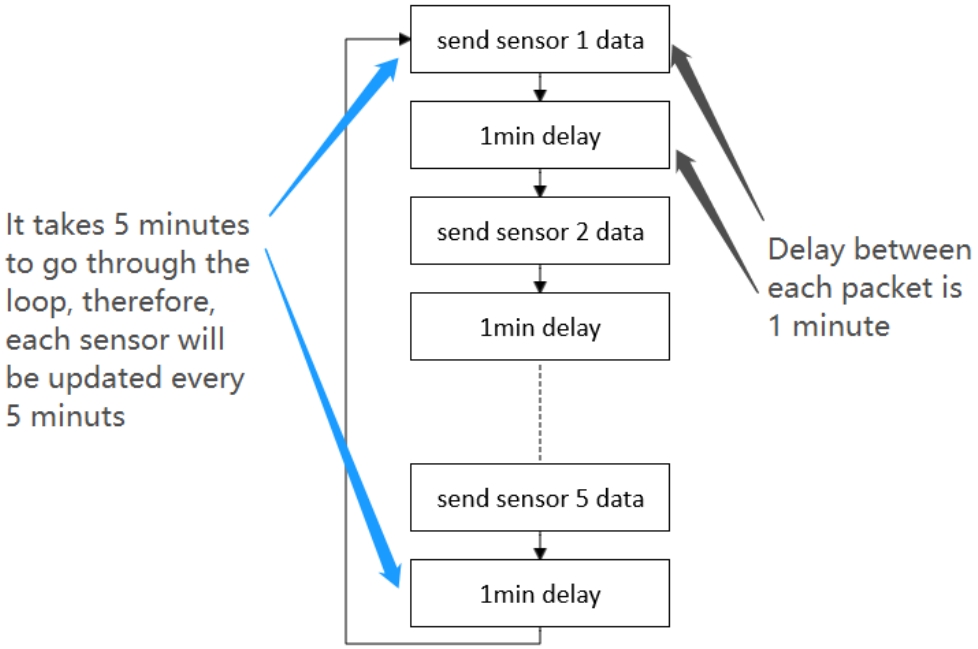


FIGURE 4.17 Sensor timing loop.

3. Mobile Sensing Nodes

Robots act as mobile sensing nodes in the WSN; they are somewhat more complicated in terms of functionality and hardware design when compared to fixed-location routing and sensing nodes.

The hardware of the mobile sensing node is built on the robot. The benefit of using a TIPoT such as a WSN development board is that Contiki OS and the ARM STM32 micro controller on board are capable of running multiple processes for both control and sensing tasks. The hardware of the cleaning and inspection robot includes a driving motor, an electrical speed controller, a servo controller, endpoint and gap sensors, a temperature sensor, TIPoT WSN mote board, and the power supply module, with the majority containing components identical to those in fixed-location routing and sensing nodes.

On top of the functionalities that the software embedded in the fixed location routing and sensing nodes provides, the software embedded in the mobile sensing nodes implements two

additional modules related to the operation of the cleaning robot. One is the drivers for all of the actuators, including the motor driver and the servo, both of which use pulse width modulation (PWM) as control signals. The other is a state machine. To control all of the robot's functions taking place during the cleaning process, and have them executed in a timely fashion and in correct sequence, a state machine functionality must be added to the Contiki operation system. This state machine uses control commands and directs endpoint sensors and gap sensors to transit among different states. Various tasks are programmed in different states for the robot to perform to complete the cleaning and detecting process.

4.6 Cloud Server

For completion of the system description, this section briefly introduces the cloud server in the SRCIS. The cloud server receives the sensing data collected by all nodes and stores them in the database. It also analyzes the data and provides data visualization. The management system running in the cloud server provides a graphic interface for site operators to control the cleaning and inspection robot locally or remotely.

4.7 Field Functionality Test

As mentioned in Section 4.1, the field functionality test of the system was conducted at a solar farm in China. The site is located in Jiangshan, Zhejiang, China, and belongs to the China General Nuclear Power Group [78]. The test site is part of a 50MW solar electric power generation plant. The main purpose of the field functionality test is to verify the functionality

of the cleaning robot and some WSN-related functionalities. First, we need to verify the control logic of the system; second, we must verify our power budget calculation; and third, we must test the remote start and cleaning status update features. The greatest problem experienced by solar electrical power generation plants using cleaning robots is the reliability of the robot. Without an accurate energy budget calculation and remote status update, some cleaning robots stop in the middle of the solar array and their status can go unnoticed for quite a while. Figure 4.18 is a satellite image depicting the plant, which includes eight rows of solar arrays. The image is taken from Google maps. We have inserted in the image the location of the static nodes and gateway along with the coverage area of each node.

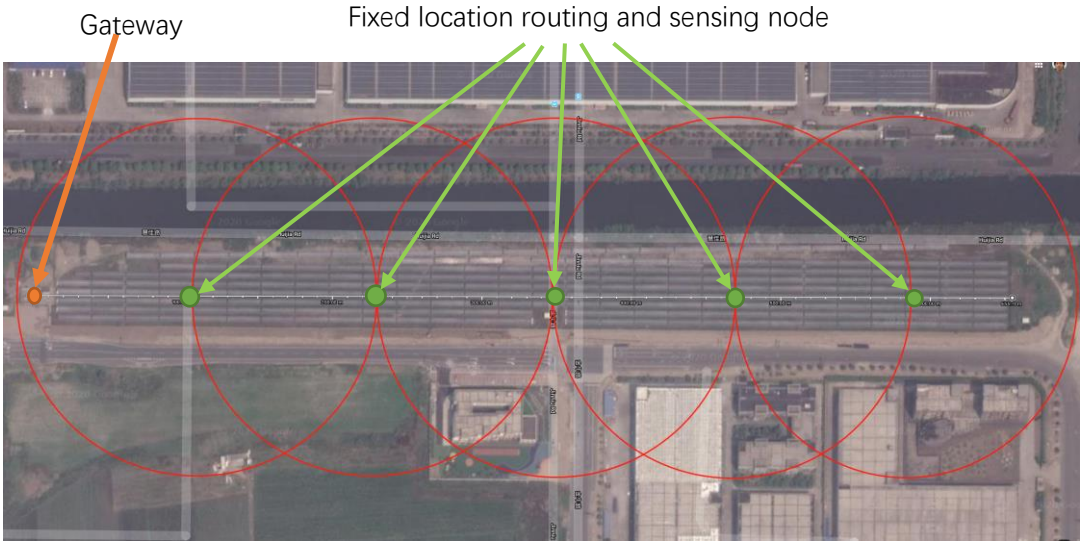


FIGURE 4.18 Satellite image of the solar panel plant used as test site.

One solar panel cleaning and inspection robot was installed on the selected solar array row. It is indicated in the red box as depicted in Figure 4.19. Six fixed-location routing and sensing nodes were placed following the long-edge direction of the field. Each is 130 meters apart from its one-hop neighbor, forming a multi-hop network with linear topology. Figure 4.20 provides three photographs taken at the test site during the experiment. The first was taken when we

were working on attaching the robot on the panel. The second illustrates part of the robot with the end-cover open. The reader can also see the solar panel that powers the robot attached to it. The third photograph was taken after installation of the robot on the solar panel row.

Figure 4.19 Test site with robot.

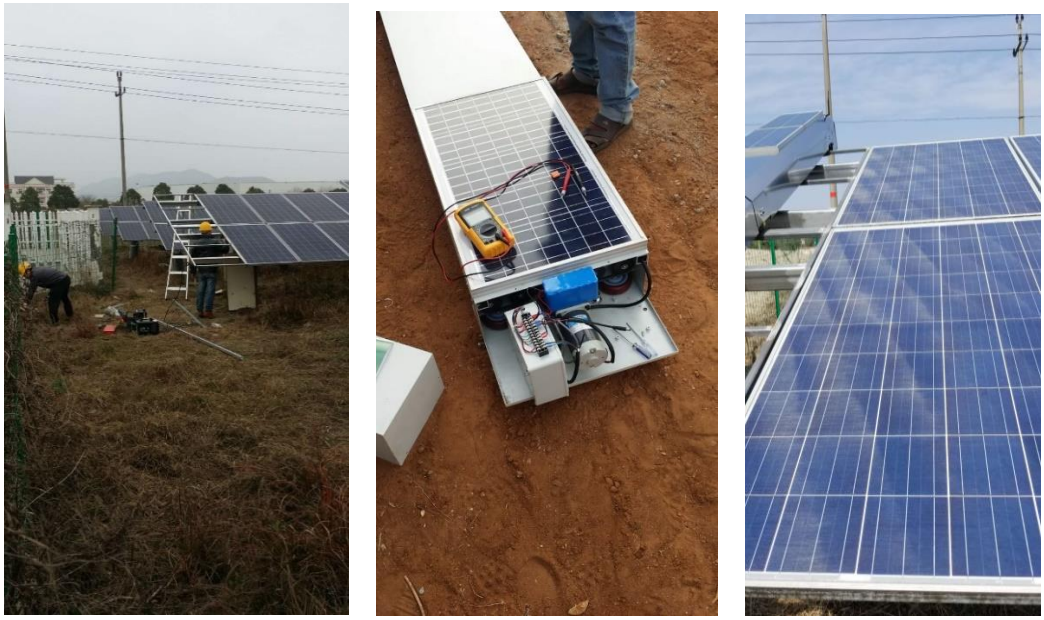


FIGURE 4.20 Photographs taken during the installation process of the robot.

The test result for the solar electric power generation plant efficiency improvement will be indicated in a comparison test. Since the sun's location and seasonal illumination power change, and the solar panel condition changes as well, it is difficult to illustrate a before and after difference with the same set of solar panels. Therefore, a different set of panels is used to collect

data demonstrating the performance in the absence or non-operation of the system we installed. One robot is deployed for cleaning a solar row on a daily basis, and five other solar rows with identical capacity were used for comparison. Since these rows are close to each other, they are exposed to the same environment: sunlight duration, dust deposits, wind, and rain. Therefore, the observed differences should be attributed to the operation of the robotic cleaning and fault detection system.

The field functionality test was conducted for a period of six months, from January 16, 2018, to June 30, 2018. Daily power generation was recorded for all six solar rows, along with the weather conditions. An example chart for April 2018 is indicated in Table 4.1. The column marked as “test row power” indicates the daily energy produced by the solar array with the robot cleaning system installed and used. The columns marked “Control 1” to “Control 5” provide the production of the other solar arrays. The robot was scheduled to make the cleaning cycle at 9 p.m. every day. The daily and statistical field test results indicate that the power generation increases with robotic cleaning. Power generation was not only recorded and compared on an aggregate basis but based on weather conditions as well: sunny/cloudy, which is indicated with red font in the table or overcast/rainy/snowy, which is in black. In the case of snowy days, the energy production difference between robotic cleaning versus non-robotic cleaning increases further in favor of the robotic cleaning operation, because snow removal by brushing takes place on a regular basis when robotic cleaning is used. At the beginning of the six-month test, the production level of the row that was later cleaned by the robot was monitored and compared to the average power production of the panels belonging to the control group. The results indicated that its production was lower.

TABLE 4.1 CGNPG Zhejiang Site Performance Test.

April Date	Weather	**Test Row** power	**Control 1** power	**Control 2** power	**Control 3** power	**Control 4** power	**Control 5** power	Average without robotic cleaning	Improvement %
1st	Sunny	215.8	209	213.9	198.4	198.5	197.8	203.52	6.03%
2nd	Overcast	80.9	77.7	79.9	70.7	71.1	71.8	74.24	8.97%
3rd	Sunny	207.8	200.2	206	190.6	190.8	190.3	195.58	6.25%
4th	Rainy	24	23.4	24.3	19.5	20.7	21.5	21.88	9.69%
5th	Rainy	36.6	35.7	37	31.8	32.7	33.6	34.16	7.14%
6th	Sunny	154	152.3	155.2	145.1	145.9	145.8	148.86	3.45%
7th	Sunny	258.4	254.8	258.2	244.7	246.4	244.9	249.8	3.44%
8th	Sunny	255.7	250.6	255.8	241.2	242.7	241.2	246.3	3.82%
9th	Sunny	242.9	238.1	241	229.4	231.2	229.5	233.84	3.87%
10th	Sunny	231.9	226.3	230.4	218.2	219.3	217.6	222.36	4.29%
11th	Cloudy	163.5	159.3	161.8	152	152.8	152.1	155.6	5.08%
12th	Overcast	77	73.8	75.2	68.8	69.3	70.1	71.44	7.78%
13th	Cloudy	112.9	109.9	112.4	105	105.9	106.2	107.88	4.65%
14th	Overcast	35.4	34.4	35.4	30.6	31.4	32.3	32.82	7.86%
15th	Sunny	188.3	184.4	187.8	177.4	178.6	178.2	181.28	3.87%
16th	Overcast	67.2	65	66.7	60.2	61	61.9	62.96	6.73%
17th	Sunny	239.8	234.5	238.2	225.2	226.6	225.3	229.96	4.28%
18th	Sunny	237.5	223.9	222.3	222.4	223.9	222.3	222.96	6.52%
19th	Sunny	236.4	221	219.5	220.5	221	219.5	220.3	7.31%
20th	Sunny	218.9	202.8	201.9	202.7	202.8	201.9	202.42	8.14%
21st	Sunny	150.1	137.3	137.7	137.2	137.3	137.7	137.44	9.21%
22nd	Cloudy	107.5	97.2	97.9	97.1	97.2	97.9	97.46	10.30%
23rd	Rainy	39.4	37.8	38.6	33.6	34.6	35.6	36.04	9.32%
24th	Rainy	45.4	44.3	45.4	39.5	40.4	41.4	42.2	7.58%
25th	Sunny	194.4	190	192.1	181.8	182.5	182.2	185.72	4.67%
26th	Sunny	182.8	178.4	180.5	170.6	170.7	170.5	174.14	4.97%
27th	Sunny	186.4	182.4	183.6	173	172.9	172.4	176.86	5.39%
28th	Sunny	212.9	207.1	208.3	198.7	199	198	202.22	5.28%
29th	Cloudy	134.3	130	130.6	123.2	123.5	123.8	126.22	6.40%
30th	Overcast	86.5	84	84.2	79.4	79.9	79.9	81.48	6.16%
Total		**4624.6**						**4377.94**	**5.63%**
Daily generation above 50KWH Sunny/cloudy		4443.8						4210.84	5.53%
Daily generation below 50KWH Overcast/Rainy		180.8						167.1	8.20%

Based on the collected results, the robot successfully performed the cleaning of solar panels, which contributed to a power generation improvement of approximately 5% when tested under the soiling condition at the Jiangshan solar electric power generation plant. The solar plant was built and began operation in 2016, thus it was only two years old when we performed the test, and all solar panels were still in good condition. We did not have a chance to test the hot-spot detection function. Operators of the solar power plant are more interested in the energy production gain, and considering that the panels are only 2 years old, there is little chance that they have developed hot-spot deficiencies. Therefore, we did not include fault detection as part of the functionality test.

Based on the data, there were only two days on which the cleaning robot canceled its scheduled tasks over the six-month (168 days, to be exact) functionality test period, which corresponds to 98% availability. When tracing the reasons for the lack of performance on these days, it was determined that the robot checked its battery level before beginning its cleaning task and decided it lacked sufficient energy to finish it. Consequently, the robot canceled the task and sent a message though the WSN to the site operator to report this event. We found that these were rainy days with low sun radiation level, therefore, the onboard battery might have not been fully charged. There were many rainy or cloudy days during the test period, and the system was able to store sufficient energy on most of them; thus, the inability to do so only on these two days should be considered the exception rather than the norm. The first conclusion that can be drawn is that the control logic of the robot is performing as designed. The robot is able to make the round trip and park at the docking station after it finishes its cleaning task and updates the cleaning status. Second, the energy budget is on the right track, at least for areas

having sunlight and weather conditions similar to those existing at the test site.[6] Additionally, the robot is able to decide correctly if should continue with or abort cleaning when it senses a shortage of energy. If the robot does decide to perform a cleaning trip, it finishes the cleaning task and goes back to the docking station every time. All the above information was transported through the WSN as per expectation, and in accordance with what is reported in Section 4.4. We also tested some unscheduled manual remote startups of the robot with no failure. Based on all of the above evidence, we conclude that the robot functioned as designed during the testing period. It made the right startup choice and did not stop halfway due to energy shortage, thereby providing a trouble-free operation.

[6] Onboard solar panel size and battery size depends on the location of the installation.

Chapter 5

Performance Improvement Techniques and Test Results

5.1 Cooja Simulator

Cooja is a Contiki OS-based WSN simulator, which not only simulates the behavior of nodes in a WSN but also simulates the Contiki OS. Unlike some pure parameter-based simulators that allow for setup of the communication signal channel, DIO interval, or other network parameters through the simulator interface, in Cooja, we must work with the Contiki source code to change the RPL configuration files or rewrite some of the source code related to the parameter that must be changed. In other words, the work to change the RPL behavior in Cooja is the same as that needed to change the firmware installed in a real node. Since the RPL realization in various OSs might differ, we use Cooja as the simulation tool to ensure that the simulation is a satisfactory representation of our design.

As indicated in Figure 3.14, the Contiki core system is separated from the bottom drivers and hardware, which means that the RPL performance is relatively independent from the selection of related hardware. Of course, the capabilities of hardware in terms of CPU speed,

bus speed, access speed, and size of memory will impact the speed of execution of tasks, however, the nature of functionality remains the same. Additionally, the difference might be imperceptible in terms of quality of offered service. For example, for MCUs running at 16MHz versus 32MHz [101], the time difference for completing the execution of an arithmetic or logic instruction is in the range of 10 nanoseconds[7] [102], which is insignificant when the DIO message transmission interval was set up at the scale of several seconds and the sensing data packet generation rate at the one-minute level.

In our work, we make use of the "sky-mote," a built-in simulated WSN node in the Cooja simulator, capable of running Contiki OS and performing RPL evaluations. A significant amount of research and analysis has been done based on the combination of Cooja and the sky-mote, for example, [97], [98], and [99].

Simulators also have their limitations. When considering energy consumption simulation, WSN simulators are more focused on the energy consumed by the communication process such as transmitting and receiving tasks. However, it is difficult to consider other energy consumption factors such as the MCU computation for actuator control and sensor energy consumptions. Our data aims to demonstrate how network-related changes affect the energy consumption; thus, they should be used as a guideline that provides insight into differences in performance rather than being treated as exact values.

5.2 Simulation Parameters

[7] Depending on the complexity of the instruction, most instructions take one clock cycle to complete. Some may take three to five clock cycles.

The simulation was conducted using the following setup and parameters listed in Table 5.1.

TABLE 5.1 Simulation parameters.

No.	Parameters	Setting
1.	Physical Layer	2.4GHz (IEEE 802.15.4)
2.	Media Access Layer	ContikiMAC, Radio Duty-Cycle, Framer
3.	Network Layer	RPL+6LoWPAN
4.	Transport Layer	UDP
5.	Packet Payload Size	80 bytes
6.	No. of Gateway Nodes	1, ID 25 Colored in Green
7.	No. of Fixed-location routing and sensing nodes	24, ID 1 to 24
8.	No. of mobile sensing node	1, ID 26 Colored in Red
9.	Simulation Duration	70 min for gateway location, 100 min with mobile sensing node
10.	Mobile sensing node speed	20 m/min
11.	Packet Rate	1 per min
12.	Transmission Range	150 m
13.	Fixed- location routing and sensing nodes grid	130 m apart horizontally, 80 m apart vertically
14.	Radio Model	Unit Disk Graph Medium (UDGM)

The simulation is an extension of the field functionality test that we conducted in Section 4.7 and focuses mainly on the solar power generation cleaning and inspection system specifications discussed in Chapter 4.

The physical layer is the IEEE 802.15.4 wireless communication standard with an operating frequency of 2.4GHz. The MAC layer for ContikiOS includes three parts: the framer, the radio duty cycle (RDC), and the ContikiMAC [95]. The framer is a collection of functions run when having to create a frame for transmission and/or parsing of received frames. The RDC layer controls the sleep time of nodes, while the MAC takes care of addressing and retransmission. The network layer for the simulation is formed by RPL combined with a

6LoWPAN, which provides IPv6 support to the WSN and routing with the RPL. The UDP is the transport layer protocol to simplify the communication procedure and reduce the load for the MCU and RF transceiver.

The packet payload size is set to 80 bytes, which is defined in the 6LoWPAN protocol as indicated in Figure 4.12. This setup represents the communication design we have implemented in our hardware. A total of 24 fixed-location routing and sensing nodes are used to form a WSN grid, numbered with identification (ID) numbers from 1 to 24. A single gateway is used and assigned the ID number of 25 in the simulated topologies; it is also colored in green in the figures. In our simulations, it is placed in two different locations—one in the left upper corner of the network deployment and the other at the center of the network. We do so to assess the impact of a gateway's location on the performance of the network. The mobility study will then be performed based on this study of gateway location, and the location that provides better performance will be used when performing the simulation. One mobile node is included and used to verify the performance with the modification of the RPL, which is discussed in detail in Section 5.4.

For the study conducted to assess the impact of the gateway location on the performance, we consider absence of node mobility in the network, and each simulation runs for 70 minutes simulation time. The network converges to its final topology within three minutes. After that, there is no network topology change. Each simulation for studying the mobile sensing node lasts 103 minutes. Based on the length of the solar row, it takes 1,000 meters/20 m/min = 50 min to cover the 1,000 meters distance. The, the round-trip time equals. 2*50 = 100 min. To ensure the topology of the static network topology converge in its final form, another three

minutes is added to the simulation. The mobile node remains immobile during the 3 first minutes, then it starts moving.

Fixed-location routing and sensing nodes are placed in an eight-by-three grid, with neighboring nodes 130 meters apart horizontally and 80 meters apart vertically (this placement is explained in detail in Section 5.3). Reliable communication between two nodes under no collision situation must be in place. Therefore, we set up the transmission coverage range of the nodes in the simulation to be 150 meters. Since the communication range of nodes is 150 meters, and two nodes are placed 130 meters apart, there is a 20-meter margin to compensate for any unforeseen sources of degradation in the actual system. Unit disk graph medium [103] is a radio model of the simulation environment; it assumes the simulation is performed in the case of free space propagation and that an omnidirectional antenna is used for each node. For UDGM, the strength (amplitude) of the received radio signal grows in an inversely proportional manner with the distance between transmitter and receiver.

There are three remarks we need to make before we move on to the simulation studies. First, in this thesis, we focus on single gateway deployment for the following reasons. We want to keep the topology of the network simple, so we can study the behavior of the static network and the switching patten of the mobile sensing node clearly; in reality, more gateways mean more wiring and infostructure construction, it will add purchasing and maintenance cost. Second, all the nodes in the WSN are equipped with an omnidirectional antenna This means the moving direction has little effect on the signal reception.

5.3 Study of Gateway Location

5.3.1 Test Field Modeling

In a typical field deployment, a set of fixed-location routing and sensing nodes is placed

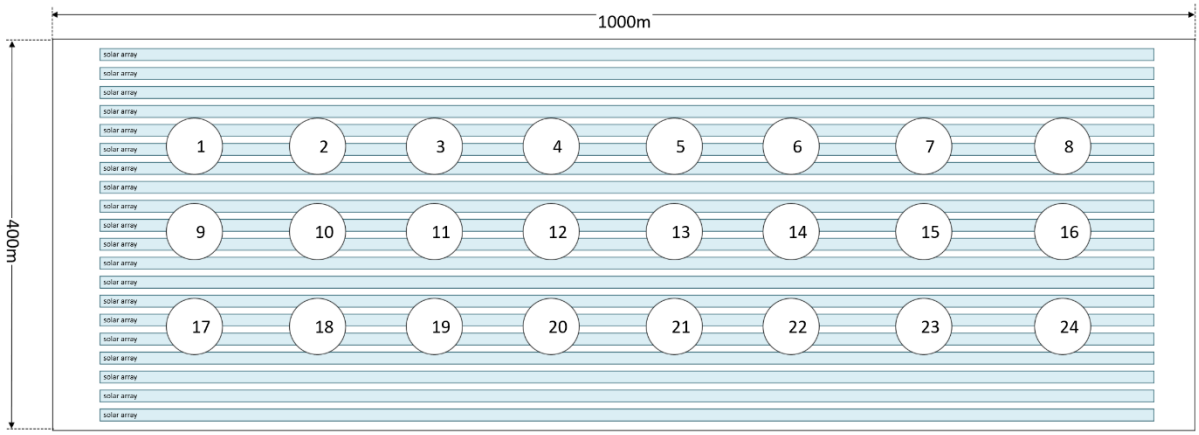
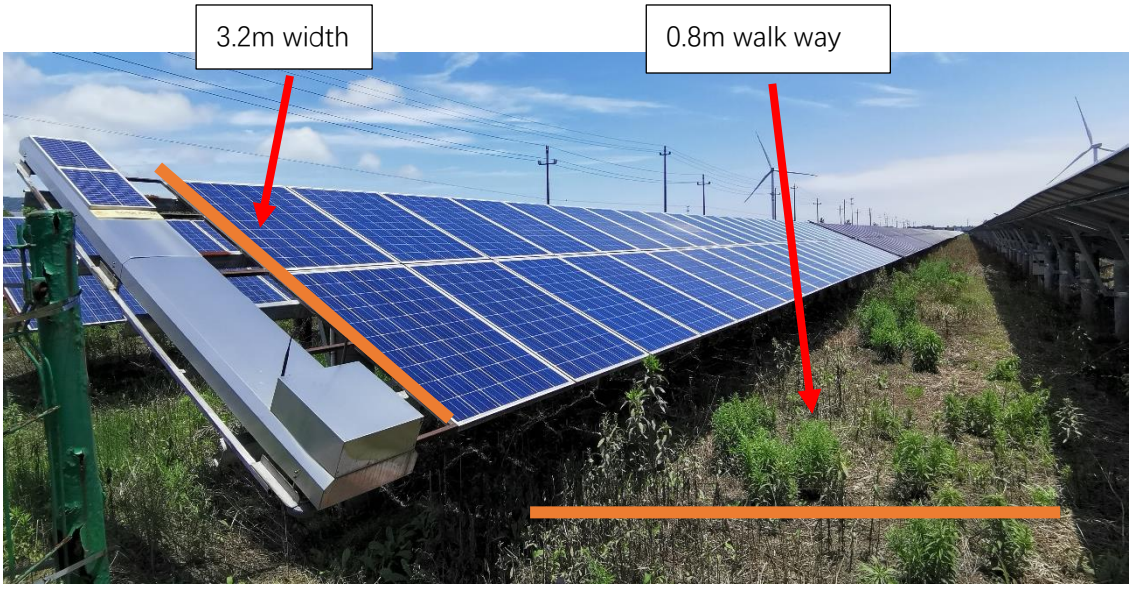


FIGURE 5.1 Test field and SRCIS wireless sensor network grid.

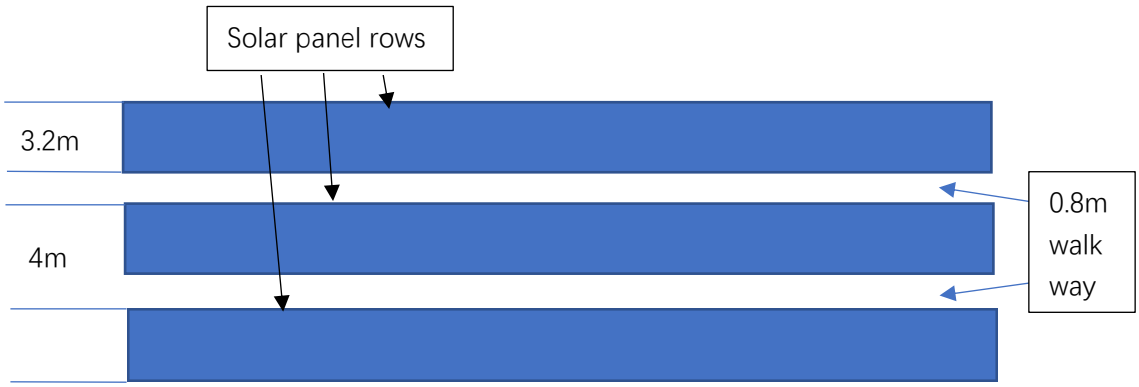
in a grid form. Each node is within the reach of its adjacent nodes, forming a multi-hop network. The grid format depends on the shape of the solar power generation plant. In our case, the test field covers a 1,000 m x 400 m rectangular-shaped area, and the fixed-location routing and sensing nodes are placed to form an eight-by-three grid, as indicated in Figure 5.1, with the horizontal light-blue color bars representing solar panel rows. This area is about the same size as the test site at Jiangshan, Zhejiang, China. It is a representative deployment for any 40MW- to 50MW-rated solar electric power generation plant.

The capacity of the plant can be calculated as follows: a 1 m-by-1.6 m solar panel is rated for 250W, which means the full load power output capacity for this panel is 250W. Figure 5.2 (a) indicates that a solar array row is formed by two rows of solar panels, the upper and the lower. This forms the width of a solar array row and is approximately 3.2 m. Then, we consider that 80% of the length of a solar array is covered by solar panels, as there should be gaps forming walkways. A 1 km-long solar array row has a capacity of 400KW considering a 0.8

m-wide walkway between each row, as seen in Figure 5.2 (b). The blue bars represent solar panel arrays, which makes the total width of a row to be 4 m. A 400-meter-wide solar electric power generation plant could have 100 solar panel array rows, which brings the total power capacity of the plant to 40MW. The coverage of the WSN is scalable and can be achieved by adding or removing fixed-location routing and sensing nodes from the grid. As mentioned earlier, the solar panel cleaning and inspection robot acts as a mobile sensing node in this network. The robot moves along the solar array across the length of the field. For example, referring to Figure 5.1, starting from node 9 and moving toward node 16, or starting from node 17 and moving toward node 24.



(a)



(b)

FIGURE 5.2 1m by 1.6m panel, 4m row width solar array row set up with robot.

5.3.2 Study of the Gateway Location

The location of the gateway node affects the topology of the network. The gateway node is assigned rank 0, which ensures that it becomes the root. No node has a lower rank. Two gateway scenarios are tested in our simulation network, which are the most distant selections. The performance was evaluated in terms of the number of hops, energy consumption, and packet loss rate (PLR).

Ten rounds of simulations were performed for each gateway location study. Each round was started "fresh" by reloading the simulation with a new random seed. A random seed is used to create randomness of the simulation by changing some of the initial conditions, such as node startup time or back-off delay. These simulations are performed with default RPL configuration parameters as follows:

- Routing metric for the OF

  The routing metric for the OF is set to be ETX, which means the rank calculation is based on the hop-distance and link quality.

- Route lifetime

The route lifetime timer is set to infinity. As mentioned in [27], this setup will provide the best stability to a static network.

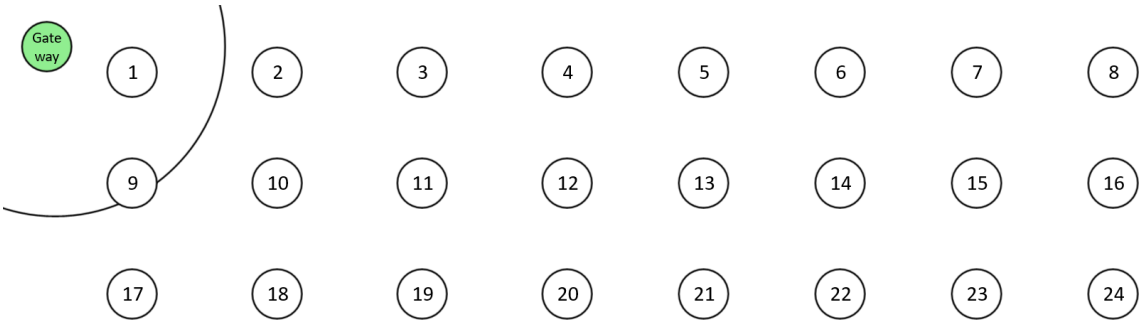
- DIO setup

  The minimum DIO interval is set to 4.096 seconds (2^{12} ms) and the maximum DIO interval is set to 1,048.576 seconds (2^{20} ms), which means the DIO interval will double if no network change has been identified after the current interval expires. Therefore, if the network does not change, the DIO interval is 4.096 seconds, 8.192 seconds, and 16.384 seconds until it reaches 1,048.576 seconds. If there is a network change related to a node, the DIO interval used by the node is reset to the minimum value, and the process is repeated.

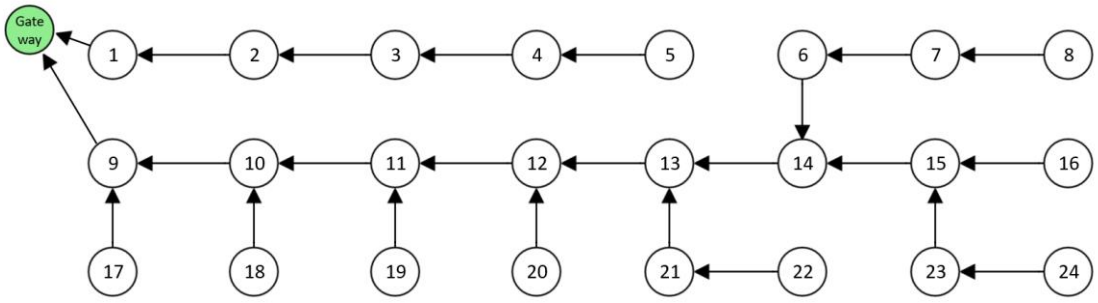
The above is the basic setting of the RPL configuration used for this simulation. The collected data are processed and used to calculate the PLR.

In one case, the gateway is placed at the location of one of the corner nodes of the parallelogram-shaped deployment area (see Figure 5.3 (a)). In the second case, it is placed in the middle of the parallelogram (see Figure 5.3 (c)).

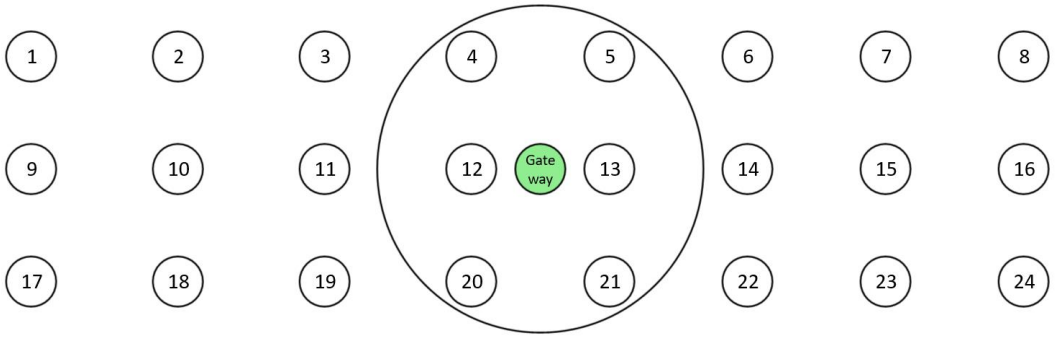
As seen by inspecting Figure 5.3 (b) and Figure 5.3 (d), the resulting network topology is quite different for the two different location placements of the gateway.



(a) Gateway at the upper left corner.

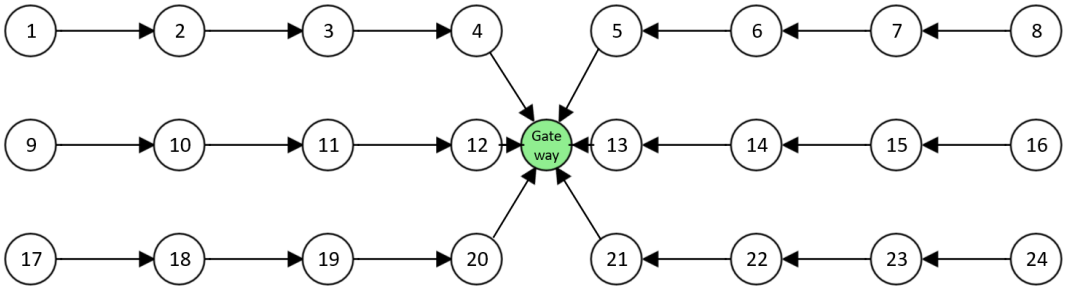


(b) The topology of the network when the gateway is at the upper left corner.



(c)

Gateway at the center



(d) Gateway in the middle of the grid.

FIGURE 5.3 Gateway locations and network topologies.

Figure 5.3 (b) and Figure 5.3 (d) are network topologies that are evident from one round of the simulation. The displayed topologies correspond to specific simulation runs. However, the topologies produced by the remainder of the simulations changed very little from those displayed. Thus, these topologies are suitable for use in the analysis of the gateway's study.

The PLR is an important network parameter that could impact the performance of an application. Figure 5.4 illustrates the PLR comparison in these two scenarios with the orange bar representing the gateway in the corner position and the blue one representing the gateway in the center position. Ten rounds of 70-minute simulations were performed, and more than 16,000 data packets were transmitted for each case. The statistical results are listed below:

- Corner gateway location

  Average PLR 0.523% with a margin of error of 0.019% and a 95% confidence interval from 0.504% to 0.542%

- Center gateway location

  Average PLR 0.073% with a margin of error of 0.013% and a 95% confidence interval from 0.060% to 0.086%

Evidently, packet losses occurring when the gateway is located at the corner are seven times higher compared to when it is located in the center. When tracing to identify the sources of lost packets, we found that most were from the leaf nodes of a long branch. The longer the path or the more the hops, the higher the chance that the packet could be lost due to collision.

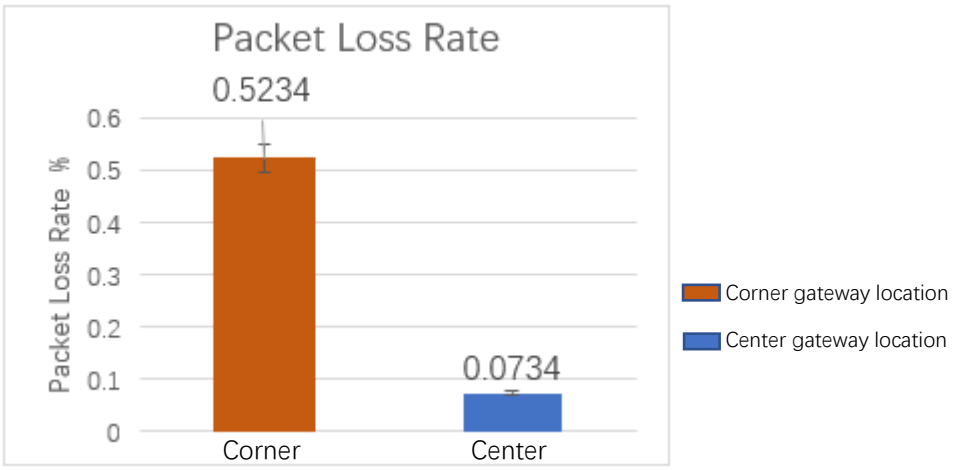


FIGURE 5.4 PLR at different gateway location.

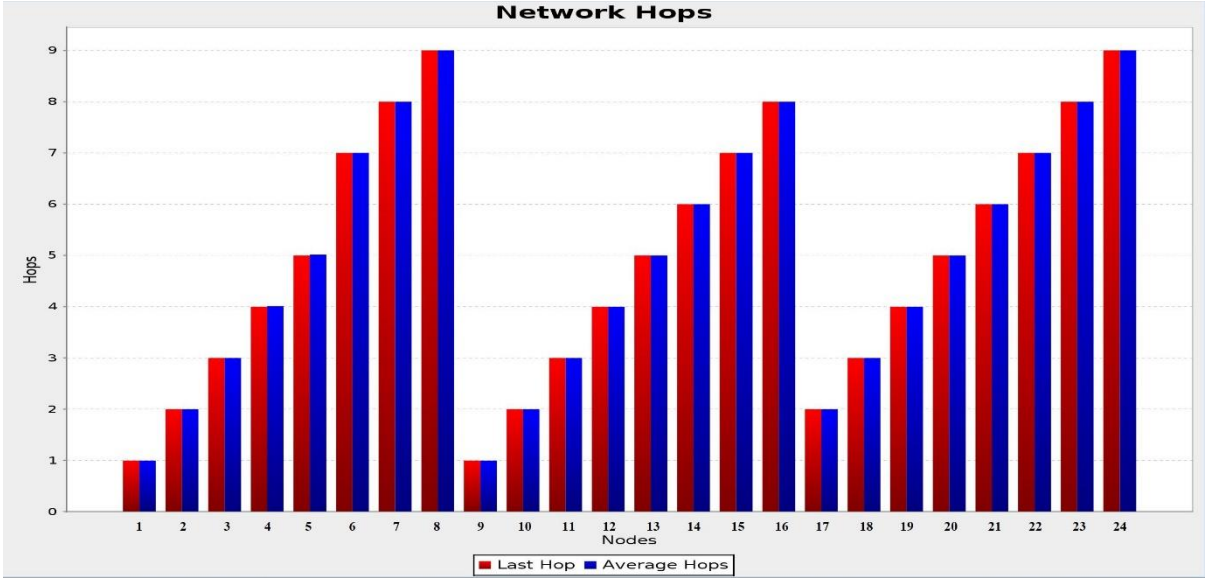
Traditionally, in solar farm deployments, it is more practical to place auxiliary facilities (e.g., control room/office, storage room and engineering facility, and accommodation facility for on-site personnel) at one of the corners of the field. The presence of the power and communication infrastructure at this location makes it practical to host the gateway at the same location. This corresponds to Figure 5.3 (a). However, with the development of technology such as power over ethernet (POE), only one cable is necessary to provide the gateway with the needed power and communication connections, which allows the gateway to be placed at additional locations other than the control room. Placement of the gateway in the center of the grid creates a more balanced topology. It is a star-like topology, having a more balanced distribution of branches and shorter routing paths. In our simulation, the number of hops for each packet traveling to the gateway was recorded for each fixed-location routing and sensing node.

A case study was performed to better understand the difference between these two gateway locations and the relevant results are illustrated in Figure 5.5. There are two data sets for each node: The red represents the final hop-distance for a node, which means the simulation has converged to its steady-state, and the blue represents the average hop-distance. For a static

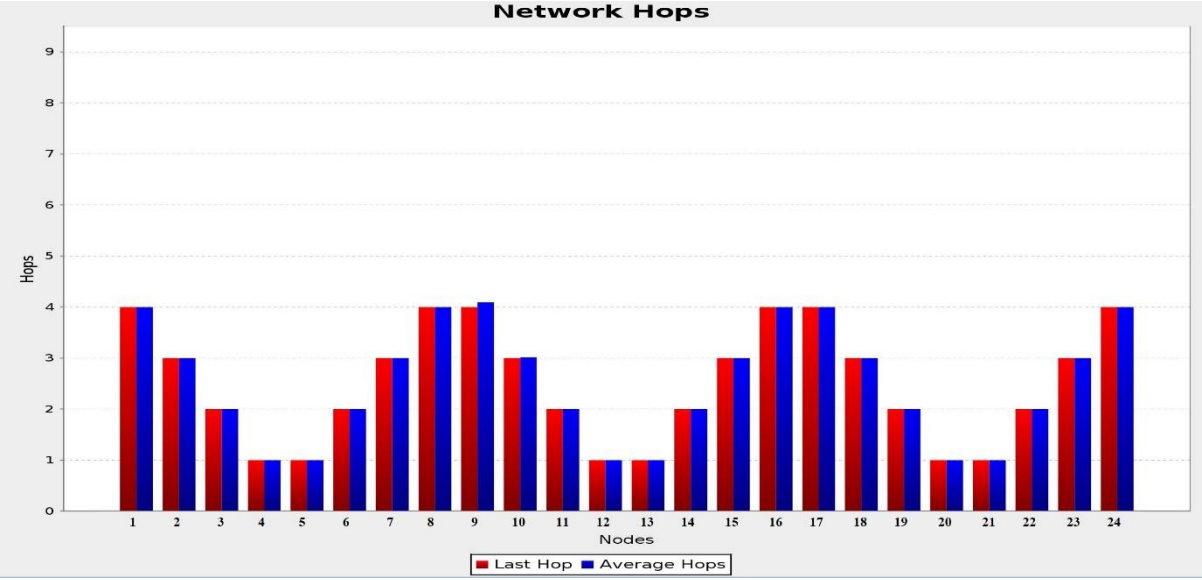
network, these two numbers are equal most of the time. However, if a node finds a better parent and switches to it for route optimization during the operation, the final hop-distance and the average hop-distance will be different. For example, in Figure 5.5 (b), node 9 has a different value between final hop-distance and average hop-distance. This is an indication of a parent switch that leads to a topology change. The average hop-distance value for node 9 in Figure 5.5 is somewhat higher than 4; this indicates that when first joining the network, node 9 has a hop-distance of 5 but not for a long time. It made the switch early, so the accumulated time for node 9 to be a hop-distance of 5 is short. If node 9 did the switch after halftime of the simulation duration, this value will be higher than 4.5. When the network forming process first started, a node is most likely to process the first DIO it has received and set the sender of that DIO as its parent node. Later, the node will receive more DIOs from different neighbors, and it may discover that new neighbors could be better parents based on rank and ETX. When that happens, the node may switch to a new parent, as long as the parent-switching threshold is met.

The set value of switching threshold is used to limit some of the unnecessary parent switching so as to keep the network stable. The parent-switching threshold is an RPL configuration parameter and is used to control the necessity for a node to switch to a different parent. This threshold is the rank difference between the current parent's rank and the potential parent's rank of a node. Based on the RPL protocol, a node in the RPL network is always looking for the lowest ranked neighbor node as its parent node. However, there is cost for a node to switch parent. When a node switches to a new parent, all of the nodes that used this node as a routing hop must change their routing paths, and those new routing paths must be updated in the gateway node. This will create more network control traffic. The parent-

switching threshold sets conditions for these parent-switching events. A smaller value of this parameter will promote more frequent parent-switching behavior; this will lead to a more optimized network topology with the cost of more network control traffic, but might make the network less stable. A larger value of this parameter will limit parent-switching events from taking place, which may lead to a less optimized network topology; however, the network tends to be more stable. The default value, 196, of the parent-switching threshold parameter set by ContikiOS is used in all simulations. To the best of our knowledge, it provides a satisfactory balance between network topology optimization and network stability.



(a) Gateway at the corner.



(b) Gateway at the center.

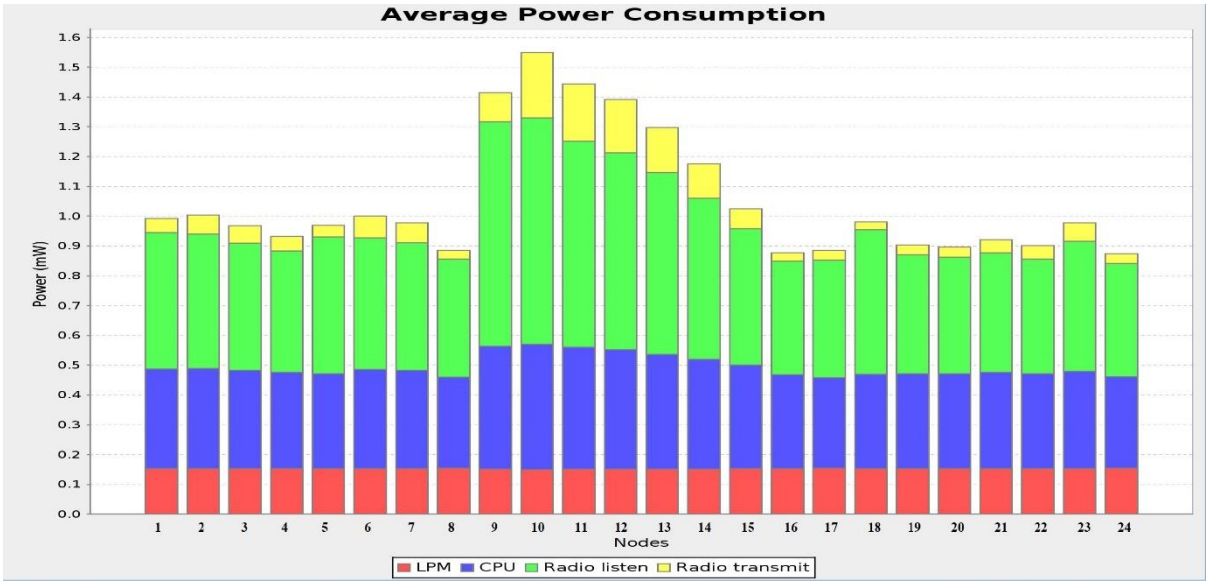
FIGURE 5.5 Gateway location and the number of hops.

By visually inspecting Figures 5.5 (a) and (b), it becomes evident that the placement of the gateway at the center results in fewer nodes having large numbers of hops. The maximum hop-count is four when the gateway is at the center, while it is nine when the gateway is placed in the corner. From Figure 5.3 (b), it can be observed that when the gateway is placed at the corner, only two nodes are connected directly to the gateway whereas six are connected directly in the other case. As illustrated in Figure 5.5 (b), nodes 4, 5, 12, 13, 20, and 21 all have a hop-distance of one.

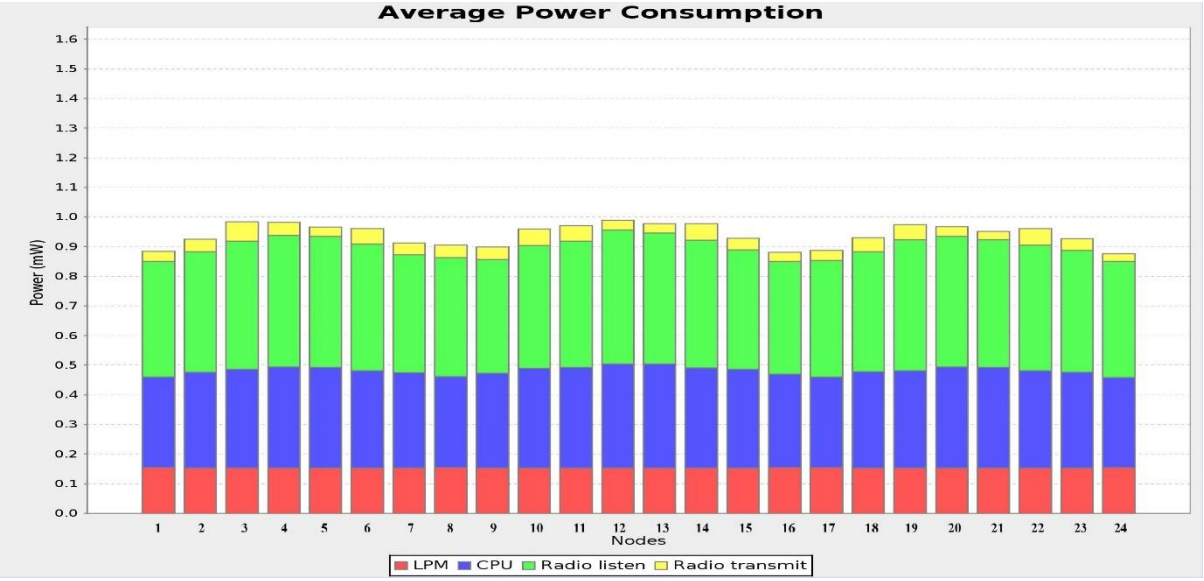
We proceed by assessing the impact of the gateway's location on energy consumption. Figure 5.6 displays the average power consumption plot captured from one of our simulation rounds. Although the results from various rounds of simulation are not the same because of the random seed, they are similar to one another and follow a noticeable pattern that resembles the one illustrated in Figure 5.6.

In Figure 5.6, the yellow part at the top of the bar represents the power consumed for

packet transmitting, the green part represents the energy used for listening and receiving, the blue part represents the energy used for CPU routing computation, and the red part is for low power mode (LPM). LPM is the energy consumption occurring when the node operates at low power mode. The nodes are running on duty cycle. When the node is not on listening and receiving cycle or transmitting cycle, it is on low power mode.



(a) Gateway at the corner.



(b) Gateway at the center.

FIGURE 5.6 Gateway locations and power consumptions.

From Figure 5.6, we can see that the first topology (Figure 5.3 (a)) creates a significantly uneven power consumption between nodes compared to the second topology (Figure 5.3 (b)). Particularly, we can see that nodes 9 and 10, which belong to the branch with the most nodes attached and are close to the gateway in terms of hop-distance, consume significantly more power than those farther away from the gateway in terms of hop-distance or located on a shorter branch. Additionally, it can be observed that node 10 has higher transmit power consumption compared to node 9, yet node 9 handles all traffic from node 10 plus the traffic from node 17. This is due to packet collision and retransmission.

Node 10 is surrounded by four nodes that send out packets. Node 9 has three neighbors, since the gateway does not send out data packets. Therefore, node 10 experiences higher traffic within its broadcast domain, which increases the collision events. Such events generate retransmissions, which consume additional energy. When the gateway is located at the corner, RPL creates fewer and longer branches in its topology. Consequently, nodes in the neighborhood of the end opposite to the side the gateway has been placed, will hop distance from the gateway that is longer to the distance they will have compared to if the gateway was placed in the middle of the deployment area. Also, the creation of less branches departing from the gateway, forces those nodes that are in communication distance with the gateway having to relay higher volume of traffic to the gateway, thus are consuming more energy as compared to what such node will be consuming if the gateway was located in the middle of the deployment area.

An additional observation we made is that the lowest energy consumption occurs at the leaf nodes in both topologies and volumes of the consumption are similar. We can use this as

our baseline to further analyze the energy consumption difference between these two gateway locations. Let us start with the case of the gateway located in the middle, as shown in Figure 5.6 (b). We know we have six evenly loaded branches, and each of the six leaf nodes have a power consumption of 0.89mW. Starting from the leaf nodes toward the gateway, we notice an increase in the energy consumption with relatively uniform incremental steps of approximately 0.025mW to 0.035mW, until it reaches the peak, which is about 0.99mW. There is only 0.1mW of power consumption difference when compared to the power consumption of the leaf node, which corresponds to 11% more power consumption compared to that of the leaf node. When the gateway is placed at the corner (Figure 5.6 (a)), the power consumption of the highest consumer, which is node 10, was recorded at 1.55mW, which is approximately 74% higher when compared to the consumption of leaf nodes. This makes the design and deployment more complicated, since the nodes close to the gateway will deplete their energy considerably faster. If the energy replenishment by means of the small PV panels used to recharge the nodes' batteries lacks enough strength to handle this disparity in energy consumption, we run the risk of having the gateway become cut off from the sensor network, in which case the entire service will fail. Because of the difference in hardware specifications, we cannot use the indicated energy consumption values directly to estimate the power usage for the TIPoT motes, however, this simulation makes evident the interrelationship between gateway location and the energy replenishment apparatus used with the nodes.

Another worthwhile observation is that while the consumption of leaf nodes when the gateway is placed in the middle is balanced (meaning leaf all nodes have close to equal energy consumption values), it is not the case for when the gateway is placed at the edge. In this case,

there are eight leaf nodes, but, as can be verified through inspection of Figure 5.6 (a), there is quite noticeable disparity among the consumption levels of the various leaf nodes. This is due to the number of neighboring nodes an edge node has and the level of traffic intensity these neighboring nodes are required to handle. Nodes 8, 16, 17, and 24 have one-hop neighbors, which themselves have low traffic volumes to process (transmit). This is because they are either leaf nodes themselves or, if they are not, they have a small number of children associated with them, for which they must act as relay. Therefore, they do not consume significant bandwidth from the wireless medium. Transmissions are less frequent, and the frequency of requiring retransmissions is low as well, since the utilization of the wireless medium within their location is low. However, this is not the case for nodes 5, 19, 20, 22, and 18. Each of these nodes has one-hop neighbors with a large number of children; such nodes must transmit more frequently. The presence of more than one such node within the hearing distance of the leaf node will also generate a higher number of failed transmissions, thus, retransmissions must occur.

From Figure 5.4 through Figure 5.6, it can be concluded that the gateway location affects the performance of the WSN. Ideally, the gateway should be deployed at a location that minimizes the hop distance, which leads to balanced power consumption and a lower PLR. This study provides a deployment guideline for the WSN gateway.

5.4 Proposed Performance Improvement Techniques for RPL Routing of Mobile Sensing Nodes

In Chapter 2, we discussed the problems rising from having to support mobile sensing

nodes. To ensure a smooth operation of the SRCIS, we proposed and implemented new techniques involving three mechanisms that address the weaknesses.

5.4.1 Test Field Setup for Mobile Sensing Node Simulation

The simulation for studying the performance improvement techniques will take place in the same test field setup as indicated in Fig. 5.7, with the gateway placed in the center of the grid, which has been demonstrated to provide better network performance. The mobile node, having been colored red in the figure below, starts at the left edge of the field. The mobile node will make a round trip, then return to its starting position. The speed of the mobile sensing node is set at 20 meter per minute with a data rate of one packet per minute. The mobile sensing node also has a signal coverage of 150 meter. Detailed settings can be found in Table 5.1.

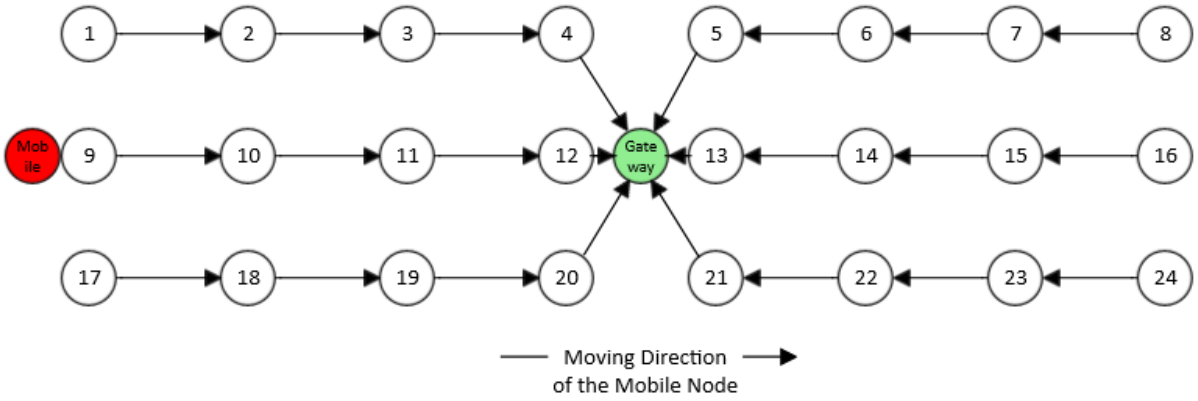


FIGURE 5.7 Mobile sensing node simulation setup

5.4.2 Mobile Sensing Node as Leaf Node

In our application, since the robot carrying the mobile sensing node is constantly moving while collecting data, the mobile sensing node was set as the leaf node; in other words, no mobile sensing nodes can have a child node. By doing so, when a mobile sensing node moves out of its parent node's coverage, only the mobile sensing node itself is affected.

It is simple to achieve this setup; we must only disable the DIO message system on the

mobile sensing node. Doing so will eliminate the DIO advertisement from a mobile sensing node. If a node is not advertising itself with a DIO, no other node will be able to use it as a parent node. Eliminating the need for advertising a DIO can also help the node to save energy.

5.4.3 DIO interval and route lifetime settings

In Chapter 2, the DIO messaging-system and trickle timer were discussed. The trickle timer is used as a dynamic scheduler for issuing DIO messages. In Contiki, the interval between two successive DIO messages is controlled by the trickle timer, which has a default minimum value (*Imin)* of 4.096 seconds and a default maximum value (*Imax)* of 1,048.576 seconds [24]. The DIO message transmission interval will start with *Imin* and double the value when the timer expires, and will be doing so until it reaches *Imax* if there is no network change. The DIO interval will remain at *Imax* until a network change is detected. The more stable the WSN is, the fewer DIO messages are needed. However, with the existence of a mobile sensing node, the network is no longer fully static, and events of loss of connectivity from the parent node are more frequent. When encountering such an event, a node will have to go through the route repair process as discussed in Section 2.5, which could take a long time to complete. The trickle timer for the DIO interval works well for stable networks but could be less effective when dealing with a network containing constantly moving nodes. The mobile nodes must receive DIOs from parent nodes frequently enough to ensure that the connections still exist for their data transmission needs, or if a connection is lost, the repair process starts within a short time after the loss.

We propose to replace the trickle timer algorithm for scheduling the DIO interval with a periodic timer. We consider this to be a better approach when having to address constantly

changing route topology due to the presence of mobile sensing nodes. In our application scenario, the fixed-location routing and sensing nodes are parent nodes to the mobile child nodes, and their DIO intervals tend to stay at *Imax* because the network is highly stable if the mobile leaf nodes are removed. The DIOs periodically sent out from the fixed-location routing and sensing nodes work like heartbeat signals, with the mobile node using the DIOs to determine if the parent node is still within its reach. It should be noted that if the DIO interval is set too short, there will be too many DIOs, which may cause network congestion, subsequent packet losses, increased power consumption, and bandwidth waste. On the other hand, if the DIO interval is set too long, the mobile node may not be able to update its route quickly enough when it moves out of the coverage of its parent and, consequently, will be cut off from the network for too long.

Please note that the objective is to have the mobile node assess that it is moving away from its current parent, coming closer to another node that can become its parent, and make the switch before it loses network connectivity. However, it is possible that the mobile node will move outside the coverage area of its parent prior to having entered the process of changing parents. A mechanism is required to be in place to enable the mobile nodes to handle this case. The route lifetime timer (RLT) exists within the RPL for this purpose. However, Contiki sets a default and very large value in this counter (for all practical purposes and for the time scale in which we operate, it is as we had set “infinity”). We change this by assigning a certain value, an action that “puts this counter into use.” When the RLT expires, the mobile node deletes its current parent from the parent list, sets the rank of itself to infinite, and acts like a new node searching for any existing DODAG and attempting to join it. Should a DIO message be

received by the mobile node before the RLT counter expires, the RLT counter is reset, and the countdown begins again. The two timers, namely, DIO message generation interval and RLT, have significant impacts on the time needed for the mobile node to handle parent loss and rejoin the network, and thus need to be fine-tuned according to the behavior of the mobile node and application requirements.

In our application scenario, the mobile node has a sensing data generation rate of one packet per minute. The mobile node moves along the solar row and enters and exits the coverage area of fixed-location routing and sensing nodes within its reception area, which consequently requires it to change its parent node from time to time. After the mobile node moves out of the coverage area of its current parent, it must discard the current route and reconnect to a new parent, so that the newly generated data packet can be successfully transmitted via the new route. If either the DIO message transmission interval or the RLT, plus the time required to acquire a new parent node, is longer than one minute, transmission of the new packet might fail. On the other hand, the RLT's value must be longer than the DIO interval to keep the route alive. The longer the RLT, the more stable the network. The DIO interval must be carefully chosen: If the DIO interval is too short, there will be too many DIOs in the network, and they will increase the energy and bandwidth consumption; if the DIO interval is too long, the update rate for the mobile node to sense the existence of its parent node will be too slow. The trickle mechanism of the DIO timer is disabled and a fixed value x is assigned to the timer for periodic DIO dissemination. Then, the value of the DIO interval is calculated as 2^x ms, as specified in [27]. When x is set to 15, we will have a DIO interval of 2^{15} ms = 32.76 seconds and when x = 14, we will have 2^{14} ms = 16.38 seconds. These two DIO interval values are

likely to be satisfactory options for our application scenario. Other values are either longer than the packet arrival interval of one minute or too short. DIOs in the WSN can also be lost for various reasons, for example, collision, noise, or interference. As discussed earlier, the RLT value will be reset after receiving a DIO. If the RLT is shorter or equal to the DIO interval, although the mobile node is still in the coverage area of the current parent, the route may still be discarded because the RLT expires before the next DIO arrival. Therefore, the RLT values must be at least a few seconds longer than the DIO interval to reduce the possibility of falsely triggering the node to leave the current parent.

Computer simulation is used to study the effectiveness of our proposed periodic DIO message transmission mechanism and route lifetime timer setting. 10 simulation rounds were conducted for each setting listed below.

1. Baseline case (default mechanism and values): DIO dissemination follows trickle timer mechanism and the route lifetime is set to be infinity. This is the default setup in Contiki.
2. Periodic DIO with interval of 32.76s and route lifetime of 48s.
3. Periodic DIO with interval of 32.76s and route lifetime of 40s.
4. Periodic DIO with interval of 16.38s and route lifetime of 40s.
5. Periodic DIO with interval of 16.38s and route lifetime of 32s.
6. Periodic DIO with interval of 16.38s and route lifetime of 28s.
7. Periodic DIO with interval of 16.38s and route lifetime of 24s.

The PLR of the mobile node is determined and comparison between the different scenarios is made. The results are shown in Fig.5.8.

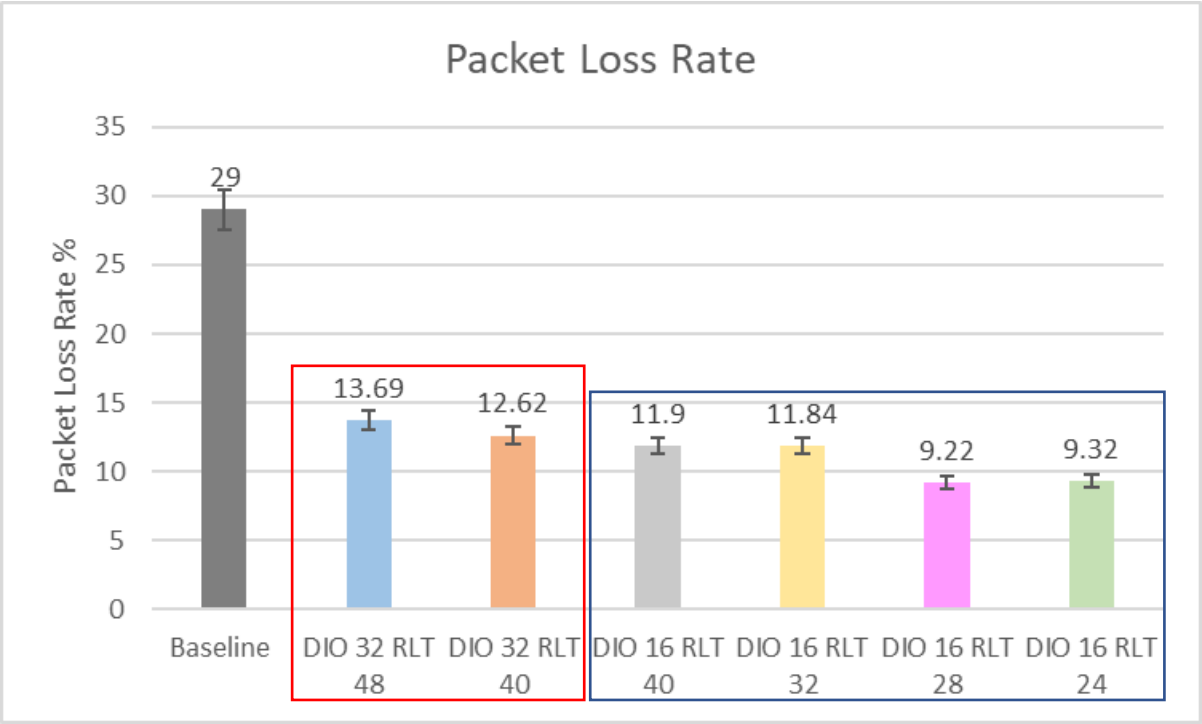


FIGURE 5.8 PLR comparison for different parameter settings.

As shown in Figure 5.8, the PLR for the baseline case is illustrated in dark grey color, 32.76s DIO interval with 48s route lifetime is illustrated in blue color, 32.76s DIO interval with 40s route lifetime is illustrated in orange, 16.38s DIO interval with 40s route lifetime is illustrated in light grey, 16.38s DIO interval with 32s route lifetime is illustrated in yellow, 16.38s DIO interval with 28s route lifetime is illustrated in pink, and 16s DIO interval with 24s route lifetime is illustrated in green. The results for 32.76s DIO interval are in the red box and the results for 16.38s DIO interval are in the blue box. The exact values produced through the simulations, used to produce Fig. 5.8, are as follows:

- Baseline, dynamic DIO interval with infinite route lifetime:

  Average PLR is 29% with a margin of error 1.23% and a 95% confidence interval from 27.77% to 30.23%.

- 32.76s static DIO interval with 48s route lifetime:

The average PLR is 13.69% with a margin of error of 2.29% and a 95% confidence interval from 11.4% to 16%.

- 32.76s static DIO interval with 40s route lifetime:

  The average PLR is 12.62% with a margin of error of 1.6% and a 95% confidence interval from 11% to 14.2%.

- 16.38s static DIO interval with 40s route lifetime:

  The average PLR is 11.9% with a margin of error of 1.58% and a 95% confidence interval from 10.4% to 13.5%.

- 16.38s static DIO interval with 32s route lifetime:

  The average PLR is 11.84% with a margin of error of 1.24% and a 95% confidence interval from 10.6% to 13.1%.

- 16.38s static DIO interval with 28s route lifetime:

  The average PLR is 9.22% with a margin of error of 0.95% and a 95% confidence interval from 8.27% to 10.2%.

- 16.38s static DIO interval with 24s route lifetime:

  The average PLR is 9.32% with a margin of error of 1.03% and a 95% confidence interval from 8.29% to 10.3%.

Compared to the baseline case, all the periodic DIO intervals with RTL value different than infinity, achieved significantly lower PLR for the mobile node. Based on the results, we can conclude that regardless if DIO is 32.76s or 16.38s, PLR decreases as route lifetime value decreases. This is because the route lifetime value affects the time a mobile node needs to wait before discarding the current parent and starting the process for a new route. This value is also

the time a node has to wait before leaving an outdated parent node in the worst-case scenario. The longer the waiting time is, the longer the time becomes for the mobile node to start the process of rejoining the network. However, shorter route lifetime comes with a cost as mentioned earlier. Among all the simulated scenarios, the lowest PLR for the mobile node was achieved with 16s DIO interval and 28s or 24s route lifetime. Further reduction of the route lifetime will not make much difference in terms of improving the PLR for mobile nodes. It is due to the fact the fixed-location routing and sensing nodes are also responsible of retransmit the mobile nodes packets. When the fixed-location nodes start to lose packets, it will affect the PLR of the mobile node.

Setting a shorter route lifetime will affect not only the mobile nodes but also the fixed-location routing and sensing nodes. Figure 5.9 illustrates the PLRs for the static network. In all simulated cases, the closer the route lifetime is set to the DIO interval, the more packet losses from fixed-location routing and sensing nodes were observed. This is most likely due to the DIO losses in the WSN. When there is a loss of DIO packet, with a short route lifetime, a node is more likely to make a wrong decision to discard the current route although it is still in the coverage of its original parent. Then, the node will rejoin the network with the same parent since its location does not change. This process could result in brief route unavailability and packet loss. From Figures 5.8 and 5.9, we conclude the best option for our application scenarios is a 16.38 seconds DIO interval with a 28 seconds route lifetime. It is also noticed that the case with a 16.38 seconds DIO interval and a 40 seconds route lifetime achieved the lowest PLR for the entire simulated network. In this case, the route lifetime is long enough for the mobile node to receive two DIOs because the DIO is sent out every 16.38 seconds This route lifetime value

cannot only handle the DIO delays but also the case when one DIO packet is lost. This is because it provides transmission of two DIO messages within a 40s period. In our application scenarios, the PLRs of mobile nodes are significantly higher than those of fixed-location routing and sensing nodes, which makes the settings that can reduce PLRs for mobile nodes more valuable. However, if other measures can be adopted to reduce the PLRs for mobile nodes to a comparable level, the route lifetime value that considers DIO losses could be given higher consideration.

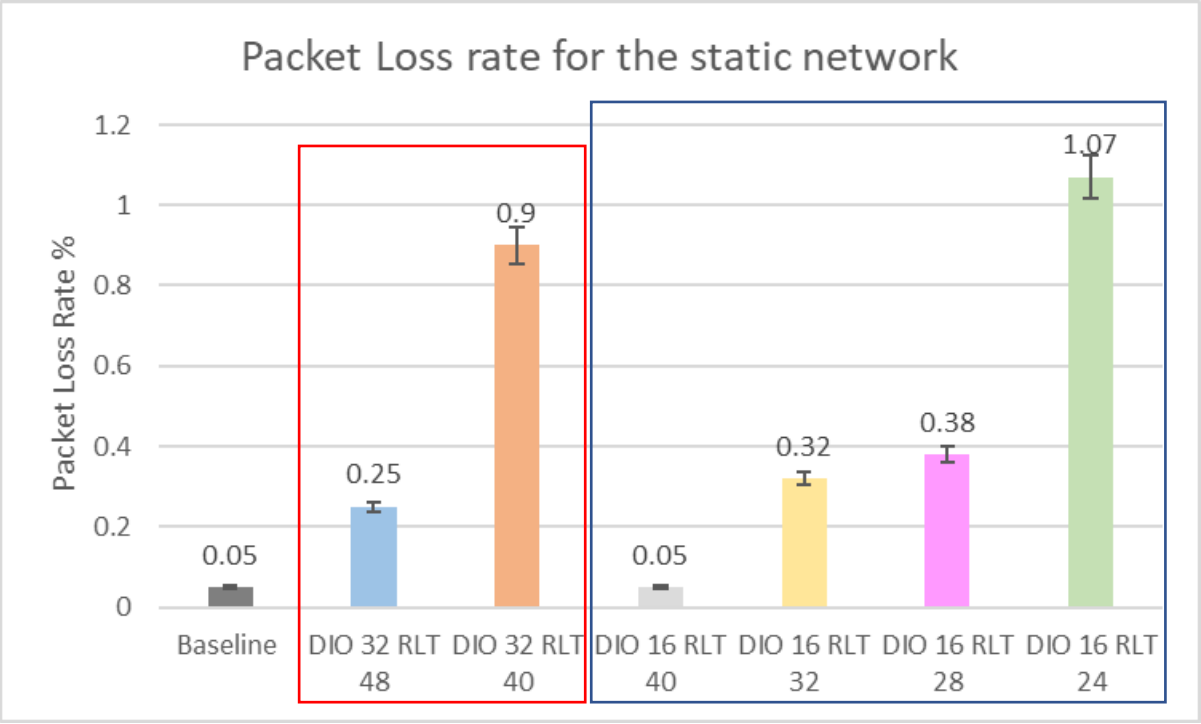


FIGURE 5.9 PLR for the static network.

5.4.4 Application Layer Redundancy

Although the PLR has been reduced to an average of 9.2% through the proposed performance improvement techniques with the 16 s static DIO interval and 28 s route lifetime parameter setting, the lost sensing data could still compromise the effectiveness of our robotic solar panel cleaning and inspection application. To further improve the reliability in data

transmission, an application layer-based method implementing data redundancy was developed and used.

As described earlier, the entire round-trip cleaning process takes about 100 minutes, and the packet generation rate is one packet per minute. A packet ID is needed to identify all of the received packets. Therefore, seven bits can identify up to 128 data packets, which is enough to cover the entire round-trip data packet identification need. The packet ID can be used to locate a defect solar panel as well. Because, the robot is moving in a constant speed of 20 meters per minute, this will give its location estimation a resolution of 20 meters. We reserved this seven-bit space in the packet at the beginning of the packet payload to identify each packet transmitted during the inspection cycle.

In our application, the sensor measures the temperature for each solar panel. The width of each solar panel is 1 meter and the robot's moving speed is 20 meters/minute. The robot takes 20 samples in one minute, which comes to one sample per panel. Each sample produces 10 bits. Depending on the situation, multiple samples can be taken on the same solar panel and the maximum or the average value of these samples will be transmitted. As a result, each panel will have one sample data transmitted and its size is 10 bits. Sensor data usually come as an integer number of bytes, but since we have the flexibility to control the structure of the payload, we will make it binary to make better use of the payload space. This adds up to 200 bits of data per minute that have to be placed in the packet's payload for transmission to the gateway.

A data redundancy mechanism is proposed, which makes use of the unused payload space for sending duplicates of data collected and sent earlier. Based on the payload size of our data packet, two previous data sets are packed into the current packet. For example, the Nth packet

can include sensor data already sent in packets N-1 and N-2. With this mechanism, even if two consecutive packets are lost, packet data can still be recovered from other packets.

The implementation of this data redundancy is straightforward. The payload size of a packet is 80 bytes, which is equal to 640 bits. As mentioned earlier, the first seven bits are reserved for packet ID. The sensing data collected two minutes ago is stored in the block of 8^{th} bit to 207^{th} bit, while the sensing data, collected one minute ago, is stored in the block of 208^{th} bit to 407^{th} bit. Finally, the most recent collected sensing data is stored from the 408^{th} bit to the 607^{th} bit. The packet payload structure is illustrated in Figure 5.16. The packet with ID = 1 does not contain actual data from the 8^{th} bit to the 407^{th} bit and packet with ID = 2 does not contain actual data from the 8^{th} bit to the 207^{th} bit.

IPv6 Header Compression 7 bytes	UDP Header Compression 1 byte	7-bit packet ID	20x10bits for N-2	20x10bits for N-1	20x10bits for N

FIGURE 5.10 The packet payload structure for data redundancy.

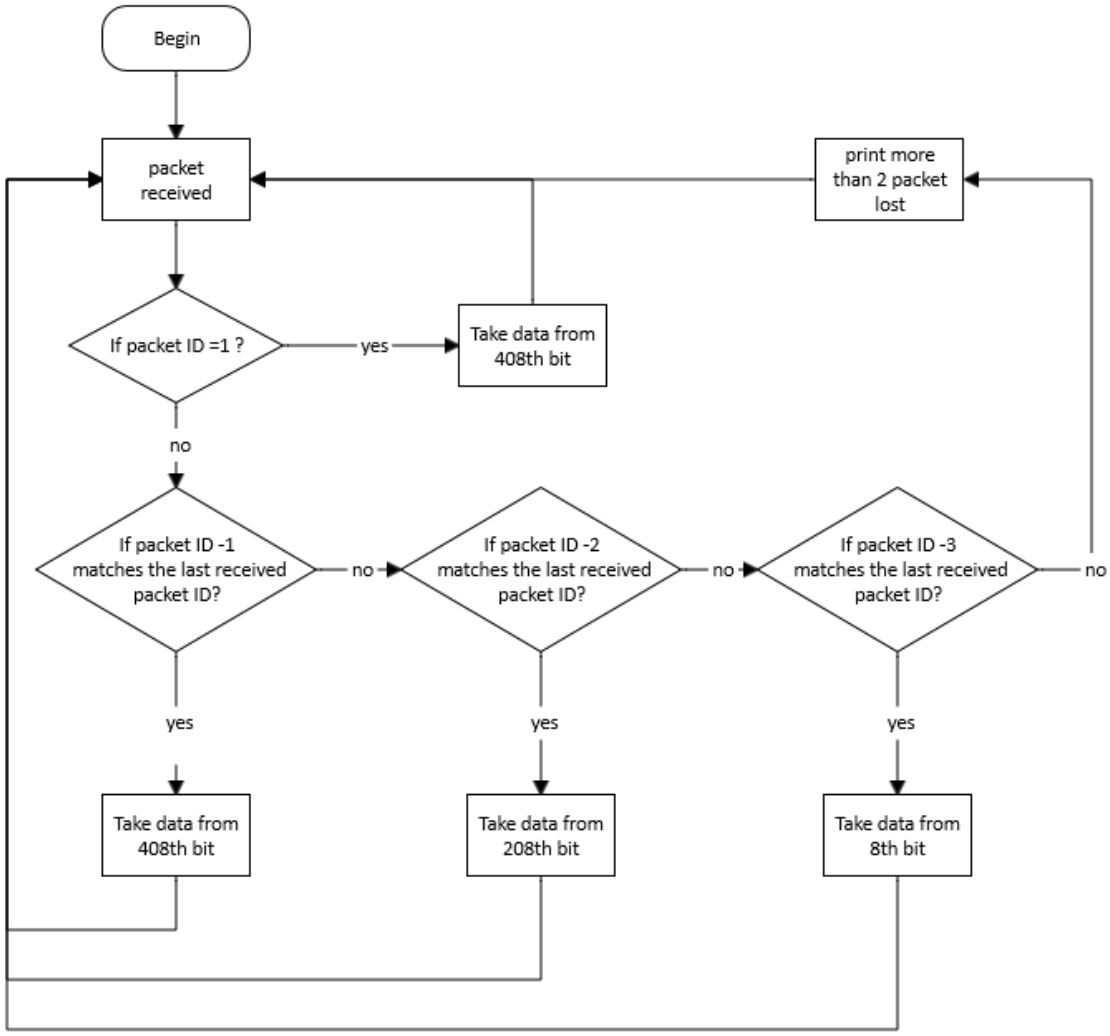


FIGURE 5.11 Using data redundancy for data receiving and recovering.

Figure 5.11 demonstrates the recovery process at the server side. When this packet is received at the server, the server checks the packet's ID. If packet ID = 1 is detected, the server will retrieve data from the 408th bit to 607th bit of the payload in the received packet, store it in the database, and then marks the last received packet ID as 1. Each time a packet arrives at the server; the packet ID is used to check for continuity. If the received packet ID-1 is equal to the last received packet ID, which indicates no packet is lost, the server will retrieve only the most recent data, located from 408th to 607th bit of the payload. If the received packet ID and the last received packet ID exhibit discontinuity, a packet has been lost. If only one packet is lost, the

server will retrieve data from the 208th to 607th bit of the payload; if two packets are lost, the server will retrieve data from the 8th to 607th bit of the payload. This procedure will allow for full recovery of the collected data if no more than two consecutive packets are lost.

To study the effectiveness of the proposed this technique, a simulation of the robot performing cleaning and monitoring cycles was carried out. For each packet loss event, we recorded the number of packets lost consequently. In the simulation, 16.38 seconds were used for DIO interval and 28 seconds for route lifetime.

As discussed earlier, the effectiveness of this redundancy mechanism depends on the frequency of occurrence of loss of consecutively transmitted packets. If more than two consecutive packets are lost, some data cannot be recovered from this redundancy mechanism. A total of 24 rounds of simulation were conducted (based on a bi-weekly schedule, this is the workload for a robot over a year) to produce the following results:

- The percentage of mobile node loss of one packet in a packet loss event is 70.15% with a margin of error of 5.2% and a 95% confidence interval from 65% to 75.4%;
- The percentage of mobile node loss of two packets in a packet loss event is 26.31% with a margin of error of 4.8% and a 95% confidence interval from 21.5% to 31.1%; and
- The percentage of mobile node loss of three packets in a packet loss event is 3.51% with a margin of error of 2.88% and a 95% confidence interval from 0.63% to 6.39%.

In Figure 5.12, it can be seen that 70% of the packet loss events are single packet losses, 26% involve two consecutive packet losses, and only 4% of the packet loss events saw a loss of three consecutive packets. This verifies that our proposed redundancy mechanism can

significantly reduce the loss of data.

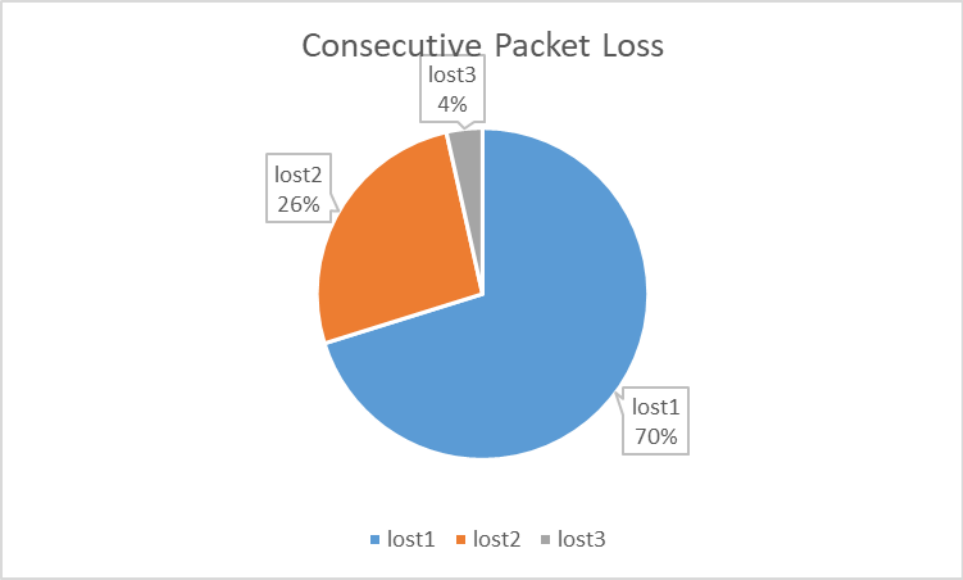


FIGURE 5.12 Consecutive packet loss

Since only the unused payload space is used, and we do not change the size or number of packets that a mobile sensing node must transmit, there should be no impact on energy consumption and network traffic. Additionally, the procedure used to decode the payload is on the server side, and the additional computing in the nodes is minimal.

Chapter 6

Conclusion and Future Research Work

6.1 Conclusion

To address the problems of power loss due to soiling and solar panel defects that most large-scale solar electric power generation plants are facing, we focused our effort in this MASc thesis research on the design and implementation of control and communication subsystems suitable for use in a SRCIS that supports autonomous cleaning and hot-spot detection.

The key modules of the designed control subsystem include the power management module, smart control module, motor and servo module, and sensors. To achieve self-sustainable functioning while controlling the overall weight of the robot, the power capacity has been carefully budgeted based on both, use scenarios and energy consumption of each component of the robot. Sensors are used for detecting solar panel defects, as well as monitoring the health of the robot and providing information to assist the control of the robot's movement. Control logic is implemented based on the cleaning and inspection requirements. Our field functionality test over a six-month period reveals that the robots with the designed control subsystem functioned reliably and performed the autonomous cleaning quite well, which resulted in an electricity production gain of more than 5%.

To ensure reliable transmission of the sensing and monitoring data between robots and gateway, we evaluated and compared the network performance when the gateway was deployed in two typical locations. When the gateway was deployed in the center of the PV deployment (and in extension center of the deployed WSN), better performance in terms of load balancing, energy consumption, and PLR was observed.

The robots move along the solar row to perform the cleaning and inspection tasks. In our performance evaluation, RPL is found to have poor performance when supporting the moving robots. To improve the performance of RPL, we proposed the implementation of several new techniques so that it can service better mobile sensing nodes. First, we declared the mobile sensing node as a leaf node, so that the location change of the mobile sensing node does not affect the rest of the network. Second, we changed the DIO scheduling mechanism and used the DIO together with our proposed new router lifetime value to ensure a timely parent search and update for mobile sensing nodes. Although the performance was improved significantly with the inclusion of these measures, packet loss still occurs at non-negligible values, especially when the mobile sensing node switches parents. To address this deficiency, we proposed and developed an application layer redundancy technique, which further protects the integrity of the collected data set. When applying this technique, we achieve drastic reduction of data loss.

The contributions of this thesis research enabled the implementation of control and communication functionalities for SRCIS able to support large-scale solar electric power generation plants. The techniques proposed for improving the RPL can be applied to other similar scenarios that involve mobile sensing nodes, which need to be switching parent nodes. While the proposed application layer-based redundancy provision technique is application-

specific—in other words, it was designed specifically for this smart application—the concept is useful to other applications where packet loss needs to be reduced, but making significant changes to the network environment is not practical solution.

6.2 Future Research Work

There is still ample room for future improvement in both robot functionality and WSN techniques. Below we comment on opportunities for future research.

To achieve a higher RoI, in many cases, solar tables or trackers need to be connected together to build longer rows. It was noticed - in our field deployments- that a structural misalignment often exists in the solar array, making it difficult for the robot to move across. This raises several challenges to our smart control system. Smarter and more accurate control algorisms needed to resolve this challenge. The robot should be able to detect obstacles, gaps, or malfunctioning trackers [75] in its cleaning and inspection path, and have the capability to take proper action to respond to these unexpected situations. Additionally, weather conditions could become hostile during the cleaning and inspection process. The robot should be equipped to adjust accordingly for safe operation or self-protection.

Most solar electric power generation plants are using the supervisory control and data acquisition (SCADA) system [76] for managing much of their existing equipment, e.g. solar tracker systems and power inverters. SCADA is an industrial standard for control and monitoring. Therefore, it is meaningful to integrate our WSN-based cleaning and inspection robot in the SCADA system. This integration will help the site operators organize the control

and monitoring tasks into a single system, having a unified interface, which will greatly simplify their daily O&M work.

While the proposed techniques achieved improved performance of our SRCIS, it should be noted that this application scenario is still in its early adoption stage. With the growing O&M needs and more complicated site conditions involving solar arrays, these techniques may require fine-tuning or modification, or new techniques may need to be developed for addressing the emerging issues. Our research has not considered the system's operation under various RF interference conditions. Some solar electric power generation plants have WiFi, Zigbee and/or other co-existing wireless networks, which could potentially interfere with the WSN that serves the SRCIS. More experiments and field tests should be conducted to address these situations, which may lead to new findings and new, more effective techniques. Also, in this research we mainly consider the uplink communication traffic (from leaf to root) with low volume of commands going through the downlink (from root to leaf). For a very large-scale network and more advanced control functionalities, more downlink traffic or peer-to-peer (P2P) communication may be required, which will have to be serviced by the WSN. For example, the control center may need to inform robots of environmental conditions or solar row status, and the robots might have to communicate with adjacent robots or trackers when a local emergency occurs (e.g., a structural failure nearby), in which case it is better for this communication to happen locally than to report to the server and then wait for a response (lowers the volume of total network traffic and reduces the end-to-end delay). It is meaningful to study P2P traffic in this case, which could also lead to the design of new technologies. Last but not least, the current application has a low data rate, which provides some advantage in applying data redundancy

for improving the reliability of data transmission. However, when the application becomes more complicated, the data rate may become much higher. Other methods will need to be developed that can support higher data rates while maintaining data transmission reliability, and do so without having to increase significantly the system's complexity.

Bibliography

[1] D. Puccinelli, M. Haenggi, “Wireless sensor networks: applications and challenges of ubiquitous sensing”, Circuits and Systems Magazine, IEEE, Vol.5, Issue 3, pp.19-21, 2005.

[2] L. Yong-Min, “The Architecture and Characteristics of Wireless Sensor network”, ICCTD '09. International Conference on Computer Technology and Development, Vol.1, pp.561 – 565, 2009.

[3] P. Rawat, K. D. Singh, H. Chaouchi, J.M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies”, The Journal of Supercomputing, Vol.68, Issue 1, pp.1-48, April 2014.

[4] V. Potdar, “Wireless Sensor Networks: A Survey”, WAINA '09. International Conference on Advanced Information Networking and Applications Workshops, pp.636 – 641, 2009.

[5] P. Gajbhiye, A. Mahajan, “A Survey of Architecture and Node deployment in Wireless Sensor Network”, Applications of Digital Information and Web Technologies, First International Conference on the, pp.426-430, 4-6 August 2008.

[6] Y. Liu, S. Wu, X. Nian, “The Architecture and Characteristics of Wireless Sensor network”, 2009 International Conference on Computer Technology and Development, pp.561-565, 13-15 Nov. 2009.

[7] [Online] Contiki Operating System, http://www.contiki-os.org/, [Accessed: Oct.2020].

[8] [Online] TinyOS, http://www.tinyos.net/, [Accessed: Oct.2020].

[9] [Online] Riot Operating System, http://www.riot-os.org/, [Accessed: Oct, 2020].

[10] S. A. Ahmed, A. V. Topalov, N. Georgiev Shakev, “A robotized wireless sensor network based on MQTT cloud computing”, 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), pp.1-6, Jun.2017.

[11] H. Jin, W. Jiang, “Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice”, IGI Global, pp.1-19, 2010.

[12] M. Aslam, S. Rea, D. Pesch, “A Vision for Wireless Sensor Networks: Hybrid Architecture, Model Framework and Service based Systems”, Digital Information Management, 2010 Fifth International Conference on, Thunder Bay, pp. 353-359, 5-8 July 2010.

[13] L. Parra, S.Sendra, J.Lloret, “Low cost wireless sensor network for salinity monitoring in mangrove forests”, SENSORS, 2014 IEEE, pp.126-129, 2-5 Nov. 2014.

[14] N. Al-Nabhan, M.Al-Rodhaan, A.Al-Dhelaan,“A distributed self-healing algorithm for virtual backbone construction and maintenance in Wireless Sensor Networks”, Wireless for

Space and Extreme Environments (WiSEE), 2013 IEEE International Conference on, pp.1-6, 7-9 Nov. 2013.

[15] T. Mishra, A.Ranjan Panda, M.Ranjan Lenka, “Energy Efficient Coverage and Connectivity with Varying Energy Level in WSN”, Computational Intelligence and Networks (CINE), 2015 International Conference on IEEE, pp.86-91, 12-32 Jan. 2015.

[16] O. Gasser, “TCP/IP communication in a WSN”, Seminar SN SS2011”, Network Architectures and Services, pp.75-82, July 2011.

[17] S. Deering, R. Hinden, RFC2460 “Internet Protocol, Version 6 (IPv6) Specification”, Dec. 1998.

[18] T.Clausen, U.Herberg, M.Philipp, “A Critical Evaluation of the IPv6 Pouting Protocol for Low Power and Lossy Networks (RPL)”, 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct. 2011.

[19] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, “RPL: IPv6 Routing Protocol for Low power and Lossy Networks”, Internet Draft, draftietf-roll-rpl-19, RCF6550, Mar.2012.

[20] J. Martocci, P. D. Mi, N. Riou, and W. Vermeylen, “Building Automation Routing Requirements in Low Power and Lossy Networks,”, Informational RFC 5867, Jun.2010.edi

[21] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks”, 1st Symposium on Networked Systems Design and Implementation, 2014.

[22] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, Standards Track RFC 4944, Sep.2007.

[23] J. Hui, P. Thubert, “Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)”, Internet Draft, draft-ietf-6lowpan-hc-15, Feb.2011.

[24] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “RFC6206: The Trickle Algorithm”, Standards Track RFC 6206, Mar.2011.

[25] T. Narten, E.Nordmark, W.Simpson, H.Soliman, “Neighbor Discovery for IP version 6 (IPv6)”, Standards Track RFC4861, Sep.2007.

[26] Belonovsky, "Development of Low-Power Device for Wireless Data Transmission under the 6LoWPAN Standard", Internet Draft, draftietf-ipv6-2461bis, 2013.

[27] F.Baker, X.Li, “Framework for IPv4/IPv6 Translation”, IETF Informational, RFC 6144, Apl. 2011

[28] T.Lee, X.Xie, L.Chang, “RSSI-Based IPv6 Routing Metrics for RPL in Lowpower and Lossy Networks”, IEEE International Conference on Systems, Man, and Cybernetics, pp.1714-1719, Oct.2014.

[29] C. Cobarzan, J. Montavont, T. No¨el, “Analysis and performance evaluation of RPL under mobility”, in: IEEE Symposium on Computers and Communications, (ISCC), Funchal, Madeira, Portugal, pp. 1–6, 2014.

[30] H. Fotouhi, D. Moreira, M. Alves, “mRPL: Boosting mobility in the internet of things, Ad Hoc Networks”, pp.17–35, 2015.

[31] M. Goyal, E. Baccelli, A. Brandt, J. Martocci, “A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network”, Standards Track RFC 6998, Aug.2013.

[32] W. Xie, M. Goyal, S. H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, “A performance analysis of Point-toPoint routing along a directed acyclic graph in low power and lossy networks, in: Proceedings of the 13th International Conference on Network-Based Information Systems” (NBiS), Takayama, Gifu, Japan, pp. 111–116, 2010.

[33] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio Ndié, Olivier Festor. “Energy-based routing metric for RPL”, [Research Report] RR-8208, INRIA. 2013, pp.14. ffhal-00779519f

[34] Lapas Pradittasnee, “A Study of the Neighbor Unreachability Detection Mechanism to Improve Performance of RPL Protocol”, International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, pp.1-6, 2016.

[35] Sheeraz A. Alvi1, A.Noor Mian, “On Route Maintenance and Recovery Mechanism of RPL”, 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp.1933-1938, 2017.

[36] V.Hoa LA, Ana R. CAVALLI, SAMOVAR, CNRS, “A misbehavior node detection algorithm for 6LoWPAN Wireless Sensor Networks”, IEEE 36th International Conference on Distributed Computing Systems Workshops, pp.49-54, 2016.

[37] [Online], Cooja Simulator, http://www.sics.se/fros/cooja.php, [Accessed: Oct.2020].

[38] W.Rukpakavong, I.Phillips, L.Guan, G.Oikonomou, “RPL Router Discovery for Supporting Energy-Efficient Transmission in Single-hop 6LoWPAN”, IEEE International Conference on Communications (ICC), pp.5721-5725, Nov.2012.

[39] [Online], IEA (2020), Solar PV, IEA, Paris https://www.iea.org/reports/solar-pv, [Accessed: Oct.2020].

[40] [Online], IEA, Solar PV power generation in the Sustainable Development Scenario, 2000-2030, IEA, Paris https://www.iea.org/data-and-statistics/charts/solar-pv-power-generation-in-the-sustainable-development-scenario-2000-2030, [Accessed: Oct.2020].

[41] [Online], IEA, Renewable power generation by technology in the Sustainable Development Scenario, 2000-2030, IEA, Paris https://www.iea.org/data-and-statistics/charts/renewable-power-generation-by-technology-in-the-sustainable-development-scenario-2000-2030, [Accessed: Oct.2020].

[42] Maghami M.R., Hizam H., Gomes C., Radzi M.A., Rezadad M.I., Hajighorbani S, “Power loss due to soiling on solar panel: A review” Renewable and Sustainable Energy Reviews, 59, pp. 1307-1316, 2016.

[43] [Online], How to find the Best solar panel angle or tilt angle, http://solarpanelsphotovoltaic.net/find-best-solar-panel-angle-tilt-angle/, [Accessed: Oct.2020].

[44] Wael Al-Kouz, Sameer Al-Dahidi, Bashar Hammad and Mohammad Al-Abed, “Modeling and Analysis Framework for Investigating the Impact of Dust and Temperature on PV Systems’ Performance and Optimum Cleaning Frequency”, Applied Scinces, April 2019

[45] [Online], Solar Power Cleaning Robot, https://us.sunpower.com/blog/2017/12/04/meet-sunpowers-new-solar-panel-cleaning-robot, [Accessed: Oct.2020].

[46] [Online], GEKKO Solar Robot, https://www.serbot.ch/en/solar-panels-cleaning/gekko-solar-robot, [Accessed: Oct.2020].

[47] [Online], Ecoppia, https://www.ecoppia.com/, [Accessed: Oct.2020].

[48] [Online], Sol-Bright, http://en.solbrighttech.com/, [Accessed: Oct.2020].

[49] Meral, M.E.; Diner, F. “A Review of the Factors Affecting Operation and Efficiency of Photovoltaic Based Electricity Generation Systems”, Renew. Sustain. Energy Rev. 2011, 15, pp.2176–2184.

[50] Skoplaki, E.; Palyvos, J.A. “On the Temperature Dependence of Photovoltaic Module Electrical Performance: A Review of Efficiency/Power Correlations”, Sol. Energy 2009, 83, pp. 614–624

[51] [Online], O&M strategies for finding module-level defects, https://www.solarpowerportal.co.uk/blogs/om_strategies_for_finding_module_level_defects, [Accessed: Oct.2020].

[52] [Online], Solar Charc, https://solarsharc.com/about/, [Accessed: Oct.2020].

[53] [Online], Thermodiagnosis of Photovoltaic Power Plants, https://www.drone-thermal-camera.com/drone-uav-thermography-inspection-photovaltaic/, [Accessed: Oct.2020].

[54] Anna Triantafillou, Panayiotis Sarigiannidis, and Thomas D. Lagkas, “Network Protocols, Schemes, and Mechanisms for Internet of Things (IoT): Features, Open Challenges, and Trends”, Wireless Communications and Mobile Computing 2018, Article ID 5349894, pp. 24 Sep 2018.

[55] N. Kushalnagar, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals”, RFC 4919, Aug.2007.

[56] Robert Heile, “802.15 WG - Wireless Personal Area Network (WPAN) Working Group”, IEEE standard, Sep. 2011. https://standards.ieee.org/standard/802_15_4-2011.html.

[57] Vinay Kumar, Sudarshan Tiwari, “Routing in IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN): A Survey”, Hindawi, 2012, article ID 316839 pp.10.58.

[58] Carels, David & Niels, Derdaele & De Poorter, Eli & Vandenberghe, Wim & Moerman, Ingrid & Demeester, Piet. (2014). “Support of multiple sinks via a virtual root for the RPL routing protocol”, EURASIP Journal on Wireless Communications and Networking. 2014. 10.1186/1687-1499-2014-91.

[59] O. Gnawali, P. Levis, “The Minimum Rank with Hysteresis Objective Function”, IETF standards track, RFC 6719, Sep.2012.

[60] JP. Vasseur, M. Kim, N. Dejean, D. Barthel, “Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks”, IETF standard track, RFC 6551, Mar. 2012.

[61] Abuarqoub, Abdelrahman & Hammoudeh, Mohammad & Alfayez, Fayez & Aldabbas, Omar. (2016). A Survey on Wireless Sensor Networks Simulation Tools and Testbeds.

[62] [Online], Tipot, https://www.tipottechnologies.com/, [Accessed: Oct.2020].

[63] [Online], J-Link Debug Probes, https://www.segger.com/products/debug-probes/j-link/, [Accessed: Oct.2020].

[64] [Online], STM32F103, https://www.st.com/en/microcontrollers-microprocessors/st

m32f103.html, [Accessed: Oct.2020].

[65][Online], AT86RF231-ZU Transceiver. Datasheet pdf. Equivalent, https://datasheetspdf.com/pdf/956069/ATMEL/AT86RF231-ZU/1, [Accessed: Oct.2020].

[66] Guerrieri, Antonio, “High-level Frameworks for the Development of Wireless Sensor Network Applications”, 10.13140/RG.2.2.32921.21600, 2012.

[67][Online], Mobility of Nodes in Cooja, https://anrg.usc.edu/contiki/index.php/Mobility_of_Nodes_in_Cooja, [Accessed: Oct.2020].

[68] [Online], IP Rated Enclosures Explained, https://www.enclosurecompany.com/ip-ratings-explained.php, [Accessed: Oct.2020].

[69] [Online], Basics of MPPT Solar Charge Controller, http://www.leonics.com/support/article2_14j/articles2_14j_en.php, [Accessed: Oct.2020].

[70] [Online], How many peak sun hours do solar panels need, https://www.solarreviews.com/blog/peak-sun-hours-explained, [Accessed: Oct.2020].

[71] Shuai Ma, Modi Jiang, Peng Tao, Chengyi Song, Jianbo Wu, Jun Wang, Tao Deng, Wen Shang, “Temperature effect and thermal impact in lithium-ion batteries: A review”, https://doi.org/10.1016/j.pnsc.2018.11.002, Nov.2018

[72] [Online], Battery Basics, https://www.progressivedyn.com/service/battery-basics/, [Accessed: Oct.2020].

[73] Thombre, Sumeet & Islam, Raihan & Andersson, Karl & Hossain, Mohammad. (2016). IP based Wireless Sensor Networks: Performance Analysis using Simulations and Experiments. Journal of Wireless Mobile Networks, Ubiquitous Computing, Dependable Applications (JoWUA). 7. 53-76.

[74] Gollu, Appala Naidu & Kumar, Jayendra & Garudachedu, Vishnu & Ramesh, Paidi. (2018). 6LoWPAN Border Router Implementation for IoT Devices on RaspberryPi. SSRN Electronic Journal. 10.2139/ssrn.3166729.

[75] [Online], Smart, Reliable Solar Trackers, https://www.nextracker.com/trackers/, [Accessed: Oct.2020].

[76] [Online], How SCADA Systems Work, https://www.elprocus.com/scada-systems-work/, [Accessed: Oct.2020].

[78] [Online], China General Nuclear Power Group, http://en.cgnp.com.cn/, [Accessed: Oct.2020].

[79] [Online], Temperature Senso, https://www.geekbuying.com/item/Waterproof-DS18B20-Temperature-Sensor-w--Adapter-Module-for-Arduino-368120.html, [Accessed: Oct.2020].

[80] [Online], Infrared Temperature Sensor, https://www.aliexpress.com/i/32840909999.html, [Accessed: Oct.2020].

[81] [Online], Wind Speed Senso, https://www.alibaba.com/product-detail/CALT-YGC-FS-0-5v-output_60675178079.html, [Accessed: Oct.2020].

[82] [Online], Wind Direction Sensor, https://www.alibaba.com/product-detail/5V-dc-0-5V-out-wind_60675789870.html?spm=a2700.galleryofferlist.0.0.1f156882tNUt2A, [Accessed: Oct.2020].

[83] [Online], Voltage Sensor, https://www.aliexpress.com/i/1972344997.html, [Accessed: Oct.2020].

[84] [Online], Current Sensor, https://www.digikey.ca/en/products/detail/allegro-microsystems/ACS770LCB-100U-PFF-T/4473982?utm_adgroup=Sensors%2C%20Transducers&utm_source=google&utm_medium=cpc&utm_campaign=Smart%20Shopping_Product_Sensors%2C%20Transducers&utm_term=&productid=4473982&gclid=CjwKCAjwkJj6BRA-EiwA0ZVPVuOKyZxTn9q2iJGABQhY6nAeYIdLcT4HYF5wa2SzNfSAko81Xbc8XRoCrBIQAvD_BwE, [Accessed: Oct.2020].

[85] [Online], Proximity Sensor, https://www.alliedelec.com/product/omron-automation/e2a-m18ls08-m1-b1/70249325/, [Accessed: Oct.2020].

[86] [Online], Proximity Sensor Internal, https://www.rfwireless-world.com/Terminology/Advantages-and-disadvantages-of-Inductive-Sensor.html, [Accessed: Oct.2020].

[87] [Online], DC Motor Drive, https://keeppower-ic.en.alibaba.com/product/60797373139-801640108/AQMH3615NS_DC_motor_drive_module_board_360W_high_power_forward_and_reverse_full_PWM_AQMH3615NS.html, [Accessed: Oct.2020].

[88] [Online], DC Motor with Gearbox, https://www.aliexpress.com/item/4000030566632.

html [Accessed: Oct.2020].

[89] [Online], Electrical Magnetic Locker, https://www.aliexpress.com/item/32817094427.

html?spm=a2g0o.detail.1000023.3.760e5b90RIMTR1, [Accessed: Oct.2020].

[90] [Online], Servo, https://www.mpja.com/download/34109MDData.pdf, [Accessed: Oct.2020].

[91] [Online], Serial Peripheral Interface, https://www.digikey.ca/en/articles/why-how-to-use-serial-peripheral-interface-simplify-connections-between-multiple-devices?utm_adgroup=General&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_RLSA&utm_term=&productid=&gclid=Cj0KCQjws536BRDTARIsANeUZ580a3agNr19BnZPA91TbTReHt5xtEzy-1n8zcObdA-8pG95HwapPMYaAoAiEALw_wcB, [Accessed: Oct.2020].

[92] [Online], Inter-Integrated Circuit, https://www.digikey.ca/en/articles/why-the-inter-integrated-circuit-bus-makes-connecting-ics-so-easy?utm_adgroup=&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_RLSA&utm_term=&productid=&gclid=Cj0KCQjws536BRDTARIsANeUZ58sVhsq12Fz9slDvwHqhS7gNNNeHyPzEKPYNy9mQA_zxzPv6T62ON0aAsapEALw_wcB, [Accessed: Oct.2020]

[93] [Online], uIP TCP/IP Protocol Stack, https://www.digikey.ca/en/articles/uip-tcp-ip-protocol-stack-demonstration, [Accessed: Oct.2020]

[94] Durvy, Mathilde & Abeillé, Julien & Wetterwald, Patrick & O'Flynn, Colin & Leverett, Blake & Gnoske, Eric & Vidales, Michael & Mulligan, Geoff & Tsiftes, Nicolas & Finne, Niclas & Dunkels, Adam. (2020). Poster Abstract: Making Sensor Networks IPv6 Ready.

[95] [Online], Rime stack, https://senstools.gforge.inria.fr/doku.php?id=os%3Acontiki, [Accessed: Oct.2020]

[96] [Online], MAC protocols in ContikiOS, https://anrg.usc.edu/contiki/index.php/MAC_

protocols_in_ContikiOS, [Accessed: Oct.2020]

[97] M. Laaouafy, F. Lakrami, O. Labouidya, N. Elkamoun and R. Iqdour, "Comparative study of localization methods in WSN using Cooja simulator," 2019 7th Mediterranean Congress of Telecommunications (CMT), Fès, Morocco, 2019, pp. 1-5, doi: 10.1109/CMT.2019.8931399.

[98] K. P. Naik and U. R. Joshi, "Performance analysis of constrained application protocol using Cooja simulator in Contiki OS," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, 2017, pp. 547-550, doi: 10.1109/ICICICT1.2017.8342622.

[99] S. N. V. Simha, R. Mathew, S. Sahoo and R. C. Biradar, "A Review of RPL Protocol Using Contiki Operating System," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), Tirunelveli, India, 2020, pp. 259-264, doi: 10.1109/ICOEI48184.2020.9142903.

[100] Z. A. Latib, A. Jamil, N. A. M. Alduais, J. Abdullah, L. H. M. Audah, and R. Alias, Strategies for a better performance of RPL under mobility in wireless sensor networks AIP Conference Proceedings 1883, 020002 (2017); https://doi.org/10.1063/1.5002020 Published 14 September 2017

[101] [Online], ARM STM 32F103 cortex-M3 spec, https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html#:~:text=STM32F103%20microcontrollers%20use%20the%20Cortex,full%2Dspeed%20interface%20and%20CAN. [Accessed: Oct.2020]

[102] [Online], ARM cortex-M3 technical reference, instruction and clock cycle, https://developer.arm.com/documentation/ddi0337/h/programmers-model/instruction-set-summary/cortex-m3-instructions, [Accessed: Oct.2020]

[103] [Online], Cooja Simulator Manual, https://www.napier.ac.uk/~/media/worktribe/output-299955/cooja-simulator-manual.pdf [Accessed: Oct.2020]