République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département d'Informatique

N° d'ordre:
N° de série:

THÈSE

Présentée pour obtenir le diplôme de

DOCTORAT EN SCIENCES EN INFORMATIQUE

THÈME

Intégration des méthodes formelles dans le développement des RCSFs

Par:
Hmidi Zohra

Soutenue le 04/10/2023, Devant le jury composé de:

Président:	Pr. Bennoui Hammadi, Professeur, Université de Biskra.
Rapporteur:	Pr. Kahloul Laid, Professeur, Université de Biskra.
Co-Rapporteur:	Pr. Benharzallah Saber, Professeur, Université de Batna 2.
Examinateur:	Pr. Chaoui Allaoua, Professeur, Université de Constantine 2.
Examinateur:	Pr. Zekri Lougmiri, Professeur, Université d'Oran 1.
Examinateur:	Dr. Ayad Soheyb, M.C.A, Université de Biskra.

Abstract

In this thesis, we have relied on formal techniques in order to first evaluate WSN protocols and then to propose solutions that meet the requirements of these networks. The thesis contributes to the modelling, analysis, design and evaluation of WSN protocols.

In this context, the thesis begins with a survey on WSN and formal verification techniques. Focusing on the MAC layer, the thesis reviews proposed MAC protocols for WSN as well as their design challenges. The dissertation then proceeds to outline the contributions of this work.

As a first proposal, we develop a stochastic generic model of the 802.11 MAC protocol for an arbitrary network topology and then perform probabilistic evaluation of the protocol using statistical model checking. Considering an alternative power source to operate WSN, energy harvesting, we move to the second proposal where a protocol designed for EH-WSN is modelled and various performance parameters are evaluated. Finally, the thesis explores mobility in WSN and proposes a new MAC protocol, named "Mobility and Energy Harvesting aware Medium Access Control (MEH-MAC)" protocol for dynamic sensor networks powered by ambient energy. The protocol is modelled and verified under several features.

Key words: Wireless Sensor Networks, Medium Access Control, Formal Modelling, Formal Verification, Statistical Model Checking, Stochastic Timed Automata, Energy Harvesting, Mobility.

Dedication

This thesis is dedicated to .....

The memory of my father,
The memory of my dear husband,
My beloved mother,
My dear brothers: Cheick and Mohamed,
My lovely kids: Meriem, Sarah and Abderrahmene,
My great supporters: Djemaa, Dalila, Latifa and Amira,

All the good people with whom God has surrounded me.

Acknowledgements

I would like to express my sincere thanks, gratitude and respect to my supervisor Pr. Kahloul Laid who has not hesitated for a single moment to direct, support and encourage me during all these years.

I would also like to thank Pr. Benharzallah Saber for his continued support.

My sincere thanks must be addressed to the members of the examination committee, Pr. Bennoui Hammadi, Pr. Chaoui Allaoua, Pr. Lougmiri Zekri and Dr. Ayad Soheyb, who honoured me by accepting the evaluation of this modest work.

A very special thanks to my teachers and colleagues in the computer science department who provided me with their help and support over these years.

Contents

List of Figures

List of Tables

GENERAL INTRODUCTION

Context and motivation

Wireless sensor networks (WSNs) have been an attractive field for several years, whether for academic research or industrial manufacturing. This type of ad-hoc networks is distributed in several critical areas: military, environmental, health and domestic. There is more and more research on several aspects of WSNs such as, deployment, localization, synchronization, security, quality of service (QoS), scalability but the research axis that has attracted the most interest from the share of the research community is that of energy savings.

As sensor nodes rely on battery with limited power, energy management is a major challenge in WSNs. Much research has been done on saving energy which is based on using hardware and software resources to minimize energy spending in order to keep the network operating for as long as possible. Energy-efficient protocols have been proposed to prolong the lifetime of the network, to the detriment of the quality of the services rendered. Another alternative for powering WSNs has emerged and is currently being actively investigated to address this challenge. It consists of extracting ambient energy (solar power, mechanical vibrations, wind, etc.) and transforming it into electrical energy to supply the sensor nodes. Evolution of energy recovery technologies has led to the development of Energy Harvesting-Wireless Sensor Networks (EH-WSNs). Each sensor node of the EH-WSNs is further equipped with energy harvesters and a storage capacitor to accumulate the recovered energy. However, the energy harvesting rates are significantly lower than the energy consumption for node operation, so that, the capacitor stores energy until it attains a certain level sufficient to operate the node. Fortunately, with storage devices having an almost unlimited number of recharge cycles, EH-WSNs can work for a long time without having to manually refill their power.

Besides this technology, advances in embedded systems have led to a new class of mobile sensor networks that improve the integration of WSNs into IoT (Internet of Things) which assumes ubiquitous detection. Therefore, mobile WSN can meet the needs of many emerging applications: the use of autonomous underwater vehicles (AUV) that travel across the ocean for search and rescue, the tracking of vital signs (heart rate, blood pressure, etc.) by wearable sensors in emergency applications, tracking objects for collecting information about their locations like as in wildlife monitoring and traffic monitoring in urban areas.

Proposing software solutions in the form of algorithms and communication protocols that support energy management is always an attractive aspect. A solution will only be accepted after it has been validated by testing, simulation, or formal verification. Simulation and testing is used to establish the correctness of communication protocols, software, and hardware. Exhaustive testing is almost always impossible with an exponential number of possibilities, thus minute errors remain unchecked and undetected until they appear at inopportune times. This lack of precision is unacceptable in critical areas, and the need for reliable verification techniques becomes essential. Formal methods can meet this need not only by verifying the properties of protocols, but also by helping us deepen our conceptual understanding of protocols.

In this context, many works [1, 2, 3, 4, 5] have attempted to validate protocols in WSNs since their first proposals. However, the rapid development of this field and the emergence of new trends of IoT, cloud and mobility are beyond the relevant formal works. Indeed, the use of traditional formal verification techniques is insufficient to overcome the increasing complexity

of these systems. The evolution of formal verification towards symbolic, statistical and distributed model checking makes it possible to overcome this complexity.

The research in this thesis is included in this context where the objective is:

- Use formal techniques to verify and evaluate the performance of existing solutions.
- Improving existing solutions.
- Propose and validate new energy-aware solutions using statistical model checking.

Main contributions

The MAC layer has attracted the interest of the research community because it is the source of various causes of energy loss such as overhearing, overhead, collision and idle listening. Therefore, the primary focus of this thesis lies on the MAC layer of WSN. On the other hand, it focuses on statistical model checking as a verification technique.

The main contributions can be summarized in the following points:

- Proposal of a stochastic generic model for the 802.11 MAC protocol for an arbitrary network topology which is independent of the number of sensors.
- Probabilistic assessment of protocol performance using statistical model checking.
- Modelling a MAC protocol designed for EH-WSNs networks using stochastic timed automata.
- Verification of the protocol using PCTL(Probabilistic Computation Tree Logic).
- Proposal of a new "Mobility and Energy Harvesting aware Medium Access Control (MEH-MAC)" protocol for dynamic sensor networks powered by ambient energy.

Thesis structure

The thesis is structured as follows:

- The first part of Chapter 1 provides an overview of the WSNs domain, focusing on their constraints, communication protocols, and their fields of application. The second part focuses on the formal verification techniques, detailing system modelling formalisms and property specification logics.
- Chapter 2 focuses on the MAC layer. It describes the communication models used in WSNs, design challenges and the main attributes used to evaluate the performance of MAC protocols. In addition, the chapter reviews MAC Protocols for WSNs.
- Chapter 3 presents the first contribution. It introduces the primary MAC scheme of the standard IEEE 802.11. It proposes a stochastic model of the protocol and a probabilistic evaluation.

- Chapter 4 concerns the second contribution. It focuses on a protocol designed for EH-WSNs. It develops the different algorithms describing the protocol. In addition, it presents the formal models developed as well as the results of the performance evaluation.
- Chapter 5 describes the third contribution. It details all aspects of the proposed protocol and presents the modelling and verification of the protocol using UPPAAL-SMC.
- General conclusion concludes the thesis, summarises the contributions and discusses issues that are open for future research.

Chapter 1

Specification and Verification of Wireless Sensor Networks

1.1 Introduction

Wireless sensor network (WSN) has emerged as one of the most promising technologies for the future. This has been enabled by advances in technology and availability of small, inexpensive, and smart sensors resulting in cost effective and easily deployable WSNs. However, the development of adequate formalisms for modelling and analysis of wireless networks has not kept pace with this. While simulation is the standard tool for analysing wireless network protocols, simulation results can depend as much on the simulator as on the design of the protocol. Formal proofs of compliance with the specifications of the application have to be provided. Applications of formal methods for the analysis of computer networks are usually motivated by a desire to study them more thoroughly, that is, to increase breadth, depth, and reliability of the analyses, thus increasing coverage, feasible system complexity, and depth of the results.

This chapter introduces existing approaches for modelling and analysis of wireless networks. Hence, it is divided into two parts. The first gives a general view on WSNs, their constraints, the different types, their fields of application as well as the different communication protocols. The second part presents the different methods of formal verification by specifying the formal modelling languages and the properties specification formalisms. Finally a conclusion that summarizes this content will be completed the chapter.

1.2 An overview of WSN

The main purpose of this section is to introduce the preliminary concepts and definitions of WSN.

1.2.1 Definition

Wireless Sensor Networks are distributed embedded systems that collect data on their environment (temperature, pressure, humidity, etc.) and relay it autonomously through their network to an end user. The network is composed of nodes, deployed in an area, that are capable of collecting data and relaying it to dedicated nodes called sinks, in a multi-hop fashion, without the need of a fixed network infrastructure. The sink node transmits the collected data to a task manager that will analyse the data and take decisions accordingly [6]. A sensor node is made up of four basic components as shown in Figure 1.1 [7]:

- **Processing unit** which executes the communication protocol's program.
- **Transceiver** controlled by the micro-controller to communicate with other nodes.
- **Sensing unit** which contains at least one sensor to collect environmental data.
- **Power unit (battery** which may be recharged by some ambient energy (solar, mechanical, etc.) collected from the environment of certain nodes.

A wireless sensor node may also have application dependent additional components such as a location finding system, a power generator and a mobilizer.

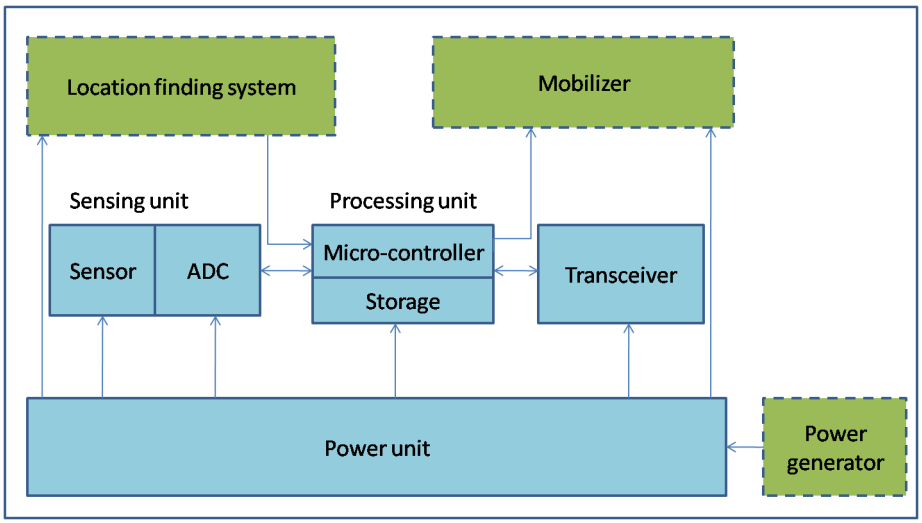


Figure 1.1: Components of a sensor node

1.2.2 Constraints and Requirements

Wireless sensor network can help human for updating data as soon as possible without moving to direct place for data collection, people do not need to reach danger place or to go to place permanently, all tasks will be done by placing wireless sensor for collecting data. But the development of these systems is a complex task for several reasons. The sensor nodes are small devices that are constrained in terms of energy resources, memory, data rate and processing capabilities. Therefore, a sensor network design is influenced by many factors [8][9]:

- **Power consumption**: due to node operation with battery, energy efficiency issue is always given to higher priority than other features in WSN. Particularly, most schemes for real-time communications are likely to choose a path with least cost repeatedly. In this case, a node's battery along the path will be quickly drained so it becomes unavailable at early time. Consequently, failure on node results in short network lifetime. So, communications protocol should be designed in energy efficient way.

- **Resource constraints**: the main material constraint is the size of the sensor which may be smaller than even a cubic centimetre which affects the processing capacity, storage and energy.

- **Fault tolerance**: the communication in WSN is unreliable due to error prone wireless medium with high bit error rates and variable-link capacity. Thus, a WSN should be reliable in order to function properly and depending on the application requirements, the sensed data should be reliably delivered to the sink node. WSNs are usually prone to unexpected node failures due to different reasons like nodes may run out of energy or might be damaged (in extreme environment conditions), or wireless communication between two nodes can be permanently interrupted. This requires WSNs to be robust to node failures. In WSN, fault tolerance can be improved through a high level of redundancy by deploying additional nodes than required if all nodes functioned properly. In case of high density deployment, sensor observations can be highly correlated in the space domain.

- **Dynamic and extreme environment conditions**: dynamic network topologies and harsh environment conditions may cause sensor node failures and performance degradation.

This requires WSN to support adaptive network operation including adaptive signal processing algorithms and communication protocols to enable end-users to cope with dynamic wireless channel conditions and varying connectivity.

- **Data redundancy**: where the sensor nodes are densely deployed in the capture field, the data captured and communicated by multiple sensors that are close of the same detected event are redundant. Data fusion and localized processing are required to address the data redundancy such that only necessary information is delivered to the end-user and communication overhead can be reduced.

- **Large scale deployment**: since WSNs may contain a large number of sensor nodes, the employed architectures and protocols must be able to scale to sizes of thousands or more. Moreover, a large scale deployment of WSN requires low-cost and small-sized sensor nodes. A WSN should be able to self-organize itself as the network topology may change due to reasons like node failure, mobility, and large scale deployments. In addition, new nodes may need to join the network, for example, to replace failed nodes, thus, a WSN must be self-reconfiguring. It can be expensive to give a unique address for each node (address-centric paradigm) especially when thousands of nodes are deployed in the application.

1.2.3 Types of Wireless Sensor Networks

Current WSNs are deployed on land, underground, and underwater. They face different challenges and constraints depending on their environment. We present hereafter five types of WSNs: terrestrial WSN, underground WSN, underwater WSN, multi-media WSN, and mobile WSN [6].

- **Terrestrial WSN**: consists of hundreds to thousands of low-cost nodes deployed in a given area, deployed either in unstructured (ad-hoc) or structured (pre-planned) manner. In this WSN, reliable communication in a dense environment is very important. Since battery power is limited, terrestrial sensor nodes can be equipped with solar cells as a secondary power source. The Energy conservation of these WSNs is achieved with multi-hop optimal routing, short transmission range, in-network data aggregation, and using low duty-cycle operations. Common applications of terrestrial WSNs are environmental sensing and monitoring, industrial monitoring, and surface explorations.

- **Underground WSN**: consists of underground sensor nodes communicated through the soil and it is used to detect and monitor underground situation. To relay information from the sensor nodes to the base station, additional sink nodes are located above the ground. The underground environment makes wireless communication a challenge due to high level of attenuation and signal loss. Underground WSNs are used in many applications such as agriculture monitoring, landscape management, underground monitoring of soil, water or mineral, and military border monitoring.

- **Underwater WSNs**: consists of sensors deployed underwater. Such nodes being expensive, only a few nodes are deployed and autonomous underwater vehicles are used to explore or gather data from them. A challenge in underwater communication is the limited bandwidth, long propagation delay, and signal fading issue. Another challenge is sensor node failure due to environmental conditions. Applications of underwater WSNs include pollution monitoring, under-sea surveillance and exploration, disaster prevention and monitoring, seismic monitoring, equipment monitoring, and underwater robotics.

- **Multi-media WSN**: consists of low cost sensor nodes equipped with cameras and microphones, deployed in a pre-planned manner to guarantee coverage. These sensor nodes interconnect with each other over a wireless connection for storing, processing, and retrieving multimedia data such as video, audio, and imaging. The challenges with the multimedia WSN include high energy consumption, high bandwidth requirements, data processing, compressing techniques and quality of service (QoS) provisioning.

- **Mobile WSN**: consists of mobile sensor nodes that can move around and interact with the physical environment. Sensor mobility occurs either when the sensor is stuck on a moving object or when the sensor is self-moving. Mobile WSN [10] is much more versatile than static WSN as the sensor nodes can be deployed in any scenario and cope with rapid topology changes. In Mobile WSN, the major environmental factors are the shared medium and varying topology. The shared medium denotes that channel access must be regulated in some way. Hence, the network topology plays a significant role in routing protocol design and also decides the transmission path of data packets to reach the desired destination. A dynamic routing algorithm must, thus, be employed unlike fixed routing in static WSN. Primary examples of mobile WSN applications are monitoring (environment, habitat, underwater), military surveillance, target tracking, search and rescue. A higher degree of coverage and connectivity can be achieved with mobile sensor nodes compared to static nodes.

1.2.4 Applications

The large emergence of WSN was originally motivated by military applications. The first wireless network resembling what we today call WSN was developed by the United States military in the 1950s to monitor Soviet submarines using acoustic sensors [11]. In 1980, the United States Defence Advanced Research Projects Agency developed a program called the distributed sensor networks [12]. Eventually governments and universities began taking an interest in WSN for applications such as air quality monitoring, forest fire detection and weather stations. At the same time, WSN made their way into industrial applications such as waste water treatment or factory automation. The decrease in size and cost of micro-sensors, the widening range of sensor variety as well as the development of wireless communications have widened the number and type of applications based on WSN technology [6][7][13].

- **Environmental monitoring**: the development of environmental monitoring system has been applied in many applications in order to assist people in their job and reduce cost and time. The applications of environmental monitoring have grown rapidly in agricultural monitoring, habitat monitoring, indoor monitoring, greenhouse monitoring, climate monitoring and forest monitoring. WSN can be useful to signal problems as when they are used to detect forest fire, flood, to control the biocomplexity of the environment and to monitor the pesticides level in the drinking water in real-time.

- **Healthcare**: an example of use of WSN for healthcare is the tracking and monitoring doctors and patients inside a hospital: each patient has small and light weight sensor nodes attached to them. Each sensor node has its specific task. For example, one sensor node may be detecting the heart rate while another is detecting the blood pressure. Doctors may also carry a sensor node, which allows other doctors to locate them within the hospital.

- **Industry**: the implementation of WSN applications in industrial automation sector is increasing the graph of productivity in marketplaces. These applications collect the real

time data acquisition, rare event detection, controlling, industrial robots, periodic data collection. Sensors are used to detect rare, random and ephemeral events such as fault detection, notification system and alarm warnings, due to uncertainty in machines. Sensors are also use for monitoring controlling the machinery and plants. The adoption of these applications decreases human errors and enhances the efficiency in production.

- **Home automation**: in this application, the sensor network is deployed in the home. The principle is that the network forms an environment, called pervasive. Smart sensor nodes and actuators can be buried in appliances, such as vacuum cleaners, micro-wave ovens, refrigerators. These sensor nodes inside the domestic devices can interact with each other and with the external network via the Internet or Satellite. They allow end users to manage home devices locally and remotely more easily.

1.2.5 Communication Protocols for WSN

In WSN, each sensor node has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructure-less architecture through the sink. The sink may communicate with the task manager node via Internet or Satellite. To assure this communication, the sink and all sensor nodes use a protocol stack which must be energy efficient and reliable in terms of communication. The model of the protocol stack given in Figure 1.2 is based on a simplification of the OSI model, it possesses 5 layers, and 3 management planes [6][14].

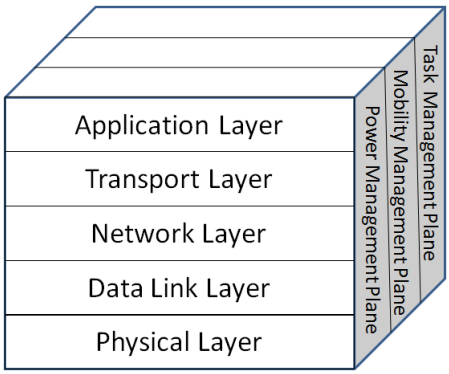


Figure 1.2: WSN protocol stack

- **Physical layer**: is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption. The physical layer design starts with the design of the radio. The design or selection of a radio is very important because the radio can impact the performance of the other protocol layers. An energy efficient radio should consume the lowest possible energy required to properly its function and communicate.

- **Data-link layer**: is used for link management, flow and error control. It is also concerned with the data transfer between two nodes that share the same link. Simultaneous transmission of data on a single channel will lead to a collision, causing loss of data and energy. To avoid this, MAC protocols are used to avoid collisions, to handle packet corruption, to minimize transmission delays and increase transmission reliability.

- **Network layer**: implements the routing mechanisms of data across the network from the source to the destination. The design of network protocols in a WSN needs to be scalable.

It should easily manage communication among many nodes and propagate sensor data to the base station. An efficient routing protocol is required at the network layer to choose a path with the minimum cost of delay, lifetime, energy or any other parameter that is more relevant to the application.

- **Transport layer**: ensures reliable data transport by inspecting the network state for congestion and reliability. Transport layer protocols in WSNs should support multiple applications, variable reliability, packet-loss recovery, and congestion control mechanism.
- **Application layer**: provides an interface to receive data from the user and to execute different types of applications.
- **Power management plane**: handles how the sensor uses its power. When the power level of the sensor node is low, the sensor node broadcasts to its neighbors that it is low in power and cannot participate in routing messages. The remaining power is reserved for sensing.
- **Mobility management plane**: detects and registers the movement of sensor nodes, so the sensor nodes can keep track of who are their neighbor sensor nodes.
- **Task management plane**: schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time.

1.2.6 Cross Layer Approach

Routing, QoS constraints, security, and time synchronization conflict directly with sensors energy consumption. So, the need that parameters must be shared among different layers of the protocol stack is necessary. For example, the QoS requirements at the application layer can be informed to the MAC layer in order to achieve better scheduling for the running application, and the channel state information can be fed to the network layer so the routing protocol can avoid paths including channels in a bad state. The cross-layer solution would involve parameters from all layers of the stack since all of them affect energy consumption to some degree. Hence, the increase in the sensors design complexity is inevitable and inversely proportional to the sensors energy capacity [14]. The cross-layered approach is more effective and energy efficient than in traditional layered approach. While traditional layered approach endures more transfer overhead, cross-layered approach minimizes these overhead by having data shared among layers. In the cross-layered approach, the protocol stack is treated as a system and not individual layers, independent of each other. Layers share information from the system. The development of various protocols and services in a cross-layered approach is optimized and improved as a whole. Various design solutions are proposed to explore the benefits of a cross-layer approach [6].

1.3 WSN Protocol Verification Methods

Formal methods are concerned with the unambiguous specification and automated validation and/or verification of software systems based on mathematical formalisms. Important validation and verification techniques based on formal methods are simulation, testing and formal verification.

- **Simulation**: is a validation technique that is concerned with some executable model of the system under consideration. A software tool called a simulator executes the model

following some scenarios (sets of possible system inputs) to determine the behaviour. This provides insight to the reactions of the system on certain inputs. The scenarios may be provided by the user or may be randomly generated. Simulation is typically useful for a quick assessment of a design, but not to show the presence of subtle errors as it is infeasible to simulate all possible scenarios. There are several established tools for the simulation of wireless networks, including NS-2, OMNet++, OPNET[15].

- **Testing**: various aspects of distributed applications such as the radio communication over a shared medium can be complex to simulate. Consequently, the results of the evaluation of a WSN system through simulations or theory can only be considered approximate. Even excellent channel models have to be confirmed by real world measurements. Due to this fact, real world experiments have increased in popularity. Experimentation on real sensor nodes is usually done in so called test-beds. A WSN test-bed is a platform for the experimentation of development projects. It allows testing the software implementation of the full protocol stack on a set of hardware nodes in a controlled environment. The packets exchanged between nodes can be traced using a sniffer. WSN test-beds enable more realistic and reliable experimentations than on WSN simulators when it comes to capturing the subtleties of the underlying hardware, software and dynamics of the WSN. The deployment of those WSN test-beds is increasing rapidly, a development that is also due to the increasing collaboration between industry and academia. WSN test-beds allow experimenting on different aspects of WSN systems conception, such as the protocol stack, resource management and network optimization. However, the development and testing of WSN systems on real platforms can quickly become tedious if the number of nodes exceeds a few dozens.

- **Formal verification**: formal verification techniques prove that a model of a system operates correctly, in contrast to testing and simulation. These techniques are based on the construction of a formal model of the system which represents the possible behaviour. The correctness requirements are stated as properties in a formal property specification language. Then it is checked whether the specification of the model contains the desired behaviour. This can be unambiguously and explicitly checked since we are dealing with formal specifications. It is important to note that formal verification techniques are only as good as the model. As opposed to testing and simulation, formal verification techniques are capable to exhaustively check the behaviour of the system under consideration.

1.3.1 Formal Verification Methods

To formally verify a system, one must model the system and its interactions in order to prove a set of properties on the model by applying formal methods. Formal methods are computer techniques based on mathematical logic that allow proving in a rigorous way that a system complies with a set of properties. The system and its properties are modelled in a mathematical language that allows establishing if the properties are verified. Numerous formal verification techniques have been proposed. The main methods are Model checking, Theorem Proving and Network Calculus.

1.3.1.1 Model Checking

Model Checking[16][17] is an automated technique that, given a finite-state model of a system and a formal property, systematically checks whether this property holds for a given

state in that model. Slightly more formally, we are interested in showing that $M \models \phi$ where M represents a Kripke structure or a labelled transition graph, as a model of system description, ϕ be a formula of temporal logic as a property. From this model of states and transitions, a semantic interpretation is generated representing all the possible behaviours of the system. Algorithms allow an exhaustive exploration of the possible states of the system. Model checking is a process that is computer aided: given M and ϕ, a computer tool called a model checker performs the check. If the property does not hold for the given model, a counterexample is provided that indicates how the error state of the model was reached. System verification using model checking goes through three phases as shown in Figure 1.3.

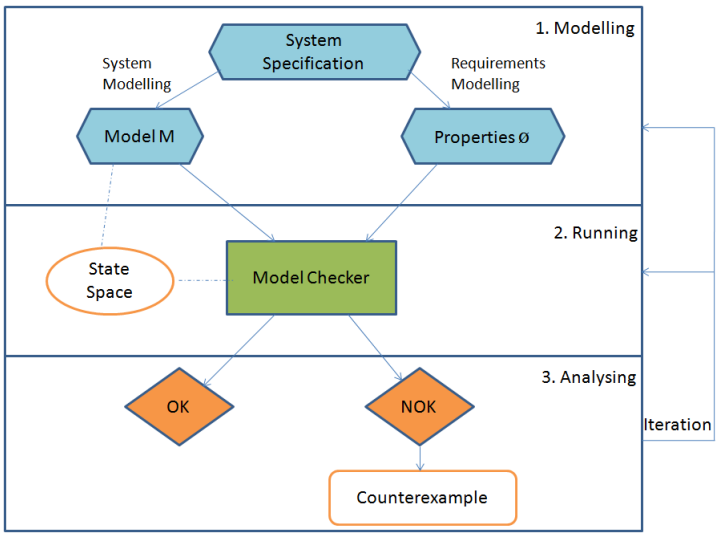


Figure 1.3: Model Checking principle

- **Modelling phase**: in which the system under consideration is modelled using the model description language of the model checker. The properties to be checked are also formalized, using the property specification language supported by the model checker.
- **Running phase**: in which the model checker systematically and exhaustively checks the validity of the given property for the constructed model.
- **Analysis phase**: in which the result of running the model checker is analysed. If the property is violated, a counterexample will be generated.

The main limitation of model checking is combinatorial explosion when the number of possible states is too big. It is then impossible to explore all of them in a reasonable time. The state explosion problem limits the application of model checking to large scale problems. Various approaches have been proposed for coping with this issue.

a) Symbolic model checking: the main insight of symbolic model checking[18] is that it is more efficient to consider large number of states simultaneously at a single step instead of traversing enumerated reachable states one at a time. Symbolic model checking facilitates such a state space traversal by allowing representations of states set and transition relations as boolean encoded formulas, BDDs(Binary Decision Diagrams), or related data structures. This allows handling of much larger designs containing hundreds of state variables. Symbolic algorithms can thus work with the FSM(Finite State Machine) represented implicitly as a formula in quantified propositional logic without the need of explicitly building a FSM graph. The first symbolic model checking tool, SMV, was developed

by McMillan in 1992 and used BDDs to combat the state explosion problem [19]. More recently, SMV has been extended and reimplemented as NuSMV and NuSMV2 [20].

b) Bounded model checking: the basic idea in BMC [21] is to search for a counterexample in executions whose length is bounded by some integer k. If no bug is found then one increases k until either a bug is found, the problem becomes intractable, or some pre-known upper bound is reached(this bound is called the Completeness Threshold of the design). The BMC problem can be efficiently reduced to a propositional satisfiability problem, and can therefore be solved by SAT methods [22] rather than BDDs. SAT procedures do not suffer from the space explosion problem of BDD-based methods. Modern SAT solvers can handle propositional satisfiability problems with hundreds of thousands of variables or more. Thus, although BMC aims at solving the same problem as traditional BDD-based symbolic model checking, it has two unique characteristics: first, the user has to provide a bound on the number of cycles that should be explored, which implies that the method is incomplete if the bound is not high enough. Second, it uses SAT techniques rather than BDDs. Experiments with this idea showed that if k is small enough(typically not more than 60 to 80 cycles, depending on the model itself and the SAT solver),it outperforms BDD-based techniques.

c) Statistical model checking: the key insight is to deduce whether or not the system satisfies the property by observing some of its executions with a monitoring procedure [23], and use hypothesis testing to infer whether the samples provide a statistical evidence for the satisfaction or violation of the specification. In contrast to a numerical approach, a simulation-based solution does not guarantee a correct result. However, it is possible to bound the probability of making an error. Simulation-based methods are known to be far less memory and time intensive than numerical ones, and are sometimes the only option. The crux of the statistical model checking approach is that since sample executions of a stochastic system are drawn according to the distribution defined by the system, they can be used to get estimates of the probability measure on executions. Starting from time bounded Probabilistic Computational Tree Logic properties, the technique has been extended to handle properties with unbounded until operators, as well as to black-box systems. Tools based on this idea have been built such as COSMOS [4], UPPAAL [24], VESTA [25] and they have been used to analyse many systems.

1.3.1.2 Theorem Proving

In the theorem proving paradigm [26] of formal verification, a deductive proof of $M \models \phi$, where M is a set of axioms and inference rules, expressed in some mathematical logic representing the system behaviour , and ϕ is also expressed in that same logic representing the system property that needs to be checked. A theorem prover is a computer tool that assists in this process of constructing a proof. Compared to a model checker, a theorem prover is less automatic since user interaction is typically required during proof construction. On the other hand, theorem provers are not restricted to finite state spaces.

1.3.1.3 Network Calculus

Network calculus [27] is a theoretical environment that allows analysing the performance of a communication network. It gives strict bounds on the system's performance, for example on hard real time constraints or QoS(Quality of Service). Communications are represented as flows going through the network. The constraints of these communications are defined by

mathematical functions called arrival curves or service curves. The calculus of performance bounds is done for a given topology, and it is not possible to represent the dynamism of the topology (node or radio links disappearing). Moreover, modelling communication protocols as service curves is based on unverified hypothesis, for example that nodes provide a minimum service.

1.3.2 Formal Modelling Languages

WSNs are wide-scale wireless multi-hop networks. To model this type of system, it is necessary to be able to represent the following aspects:

- **Concurrency**, nodes working simultaneously;
- **Communications**, neighbouring nodes in the topology can communicate between them and interfere with each other due to the broadcast nature of the radio medium;
- **Data**, we need to have data structures to represent system information (information contained in packets, topology network, etc.);
- **Probabilities**, the radio links are unreliable, so the communications are probabilistic, it is necessary to be able to express the probabilities of reception and packet loss;
- **Time**, we have to express temporal behaviours of protocols, for example the time between two events: sending and receiving a packet.

We're interested in this thesis to the formalisms that can be used to model WSN and which make it possible then to carry out a verification by Model Checking. In the following sections, we present several possible formalisms.

1.3.2.1 Petri Nets

Petri nets have been commonly used for representing concurrent network protocols [28]. Formally a Petri net is a 5-tuple $(P, T, F, W, M0)$ such that:

- $P = p_1, p_2, \ldots, p_n$ is a finite set of **places** which are marked with tokens,
- $T = t_1, t_2, \ldots\ldots t_m$ is a finite set of **transitions**,
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of oriented **arcs** that connect places to transitions and transitions to places,
- $W : F \to N$ is a function that assigns a **weight** to each arc,
- $M_0 : P \to N$ which assigns an **initial marking** to places: marking is an integer associated with a place, it corresponds to the number of tokens contained in the place.

The dynamics of the model are represented by the movement of tokens, which is described by the transition firing rules. A transition is said to be enabled by a given marking if all its input places have at least one token for each input arc from the place to the transition. That is, transition t_j is enabled by the marking M if and only if $M(P_i) > W(P_i, t_j)$, for all $P_i \in P$. An enabled transition can fire, removing tokens from input places and creating tokens in output places. If transition t_j fires, then the new marking, say M', is given by $M'(P_i) = M(P_i) + W(t_j, P_i) - W(P_i, t_j)$, for all $P_i \in P$. A temporal extension of Petri nets allows an explicit

representation of time [29]. Temporal Petri nets add time constraints to cross transitions. These constraints have the form of time intervals during which it is possible to take the transition. There is also a stochastic temporal version of Petri networks that can represent probabilities [30].

1.3.2.2 Process Algebra

Process algebra is a widely used formalism for specifying and verifying distributed concurrent software systems. In process algebra, a system is specified in the provides syntax, and the composed system is then verified against the desired properties axiomatically [31]. The system is represented in the form of a process term, using the basic operators, the communication operators, and recursion. The process terms represent the states in this labelled transition system model, and transitions correspond to actions. The resulting process term is manipulated by means of equational logic, to prove that its graph conforms with the desired external behaviour. This framework can be used to detect undesirable properties and to formally derive desirable properties of a system specification.

1.3.2.3 Timed Automata

Timed automata [32] are widely used to model and analyse the behaviours of real time systems. A timed automaton is a finite state machine with a set of clocks to ensure adherence to strict timing constraints, such as execution times, response times and communication delays. The simplest form of a constraint compares a clock value with a time constant. Timed automata only allow boolean combinations of simple constraints, i.e., for a set X of clock variables, the set $\zeta(X)$ of clock constraints δ is defined inductively by $\delta := x \leq n | n \leq x | \neg\delta | \delta_1 \wedge \delta_2$ where x is a clock in X and n is an integer constant. These clock variables are initiated with zero when the system is started, and then increase synchronously with the same rate. Clock variables can also be attached to locations as invariants. A location can be entered and stayed in only when all of its invariants are true. A formal definition of timed automaton [33]is a tuple $(Loc, Loc_0, \Sigma, X, \rightarrow, Inv, L)$ where:

- Loc is a finite set of **locations**,
- $Loc_0 \subseteq Loc$ is a set of **initial locations**,
- Σ is a set of **actions**,
- X is a finite set of **clocks**,
- $\rightarrow \subseteq Loc \times \zeta(X) \times \Sigma \times 2^X \times Loc$ is the set of **edges** with $\zeta(X)$ assigns a clock constraint named **guard** to edges,
- $Inv : Loc \rightarrow \zeta(X)$ assigns **invariants** to locations,
- $L : Loc \rightarrow 2^{AP}$ is a **labelling** function states with AP a set of atomic proposals.

In one state, time can flow as long as the invariant is true. Any transition active since this state can be taken, a transition is active if the constraint on clocks (guard) is verified. The transition is selected in a non deterministic among the active transitions. When the transition is taken, a subset clocks (2^X) is reset to 0.

Timed Automata are used to represent the temporal behaviour of systems, however, it is more difficult to represent communications and concurrency with timed automata because the

synchronization of several elements of the system would have to be encoded. Because the number of possible combination between the actions of the different elements increases in a quadratic way with the number of elements, this encoding becomes quickly complicated and long as the system's size increases. Consequently, to represent concurrency and communications, Networks of Timed Automata (NTA) have been introduced [22]. A NTA is composed of several TA working in parallel and synchronizing over common actions. NTA enable modelling time, concurrency, as well as communications through synchronization. Formally, an NTA is the parallel composition of $\{A_i\}_{1 \leq i \leq n}$ where $A_i = (Loc_i, Loc_i^0, \Sigma_i, X_i, \rightarrow_i, Inv_i, L_i)$ for $1 \leq i \leq n$.

The theory of timed automata can be used to prove the correctness of real time systems. Generally, two types of properties, liveness and safety, are concerned. As checking liveness is computationally expensive, the main effort of verifying a timed system focuses on checking the safety properties, which can be checked using reachability analysis by traversing the state-space of timed automata. It is proved that the reachability verification of timed automata is decidable, still quite expensive though.

1.3.2.4 Stochastic Extensions of Timed Automata

a) **Probabilistic Timed Automata**

Probabilistic timed automata [34] are a modelling formalism for distributed systems that support dense time, non-determinism, and probabilistic choice. They represent an extension of timed automata, for which discrete probability distributions range over the edges of the control graph.

Formally, a probabilistic timed automaton is a tuple $(Loc, Loc_0, \Sigma, X, prob, Inv, L)$ where:

- Loc is a finite set of **locations**,
- $Loc_0 \in Loc$ is the **initial location**,
- Σ is a finite set of events, such that $Act_u \subseteq Act$ are urgent,
- X is a finite set of **clocks**,
- $prob \subseteq Loc \times \zeta(X) \times \Sigma \times Dist(2^X \times Loc)$ is the **probabilistic** transition relation. $Dist(2^X \times Loc)$ denotes the couple: (probability intensity over the transition, the target location).,
- $Inv : Loc \rightarrow \zeta(X)$ is the **invariant** function associated for each location,
- $L : Loc \rightarrow 2^{AP}$ is a **labelling** function states with AP a set of atomic proposals.

A state of a probabilistic timed automaton is a pair (l, v) where $l \in Loc$ and $v \in \mathbb{T}^X$ (a clock valuation) are such that $v \lhd inv(l)$ (i.e. v satisfies $Inv(l)$). The model starts in the initial location Loc_0 with all clocks set to 0, and hence the initial state is $(Loc_0, 0)$. In each location, there is a non-deterministic choice between two types of transitions: Delay transitions which correspond to the elapsing of time in a location. They are permitted as long as the invariant condition is satisfied and no urgent transitions are enabled. Discrete transition correspond to the execution of probabilistic transitions $(l, g, \sigma, p) \in prob$. If the current location l satisfies the clock constraint g and the current event is σ, then $p(X, l')$ is the probability of resetting all clocks in X to 0 and moving to the location l'.

The semantics of probabilistic timed automata is defined in terms of timed probabilistic systems, which exhibit timed, non-deterministic and probabilistic behaviour. They are a variant of Markov decision processes and Segala's probabilistic timed automata [35].

Moreover, it is often useful to define complex systems as the parallel composition of a number of interacting sub-components. Let $PTA_i = (Loc_i, Loc_i^0, \Sigma_i, X_i, prob_i, Inv_i, L_i)$ for $i \in 1, 2$. The parallel composition of two probabilistic timed automata PTA_1 and PTA_2 is the probabilistic timed automaton $PTA_1 \parallel PTA_2 = (Loc_1 \times Loc_2, (Loc_1^0, Loc_2^0), \Sigma_1 \cup \Sigma_2, X_1 \cup X_2, prob, Inv, L_1 \cup L_2)$ where $Inv(l, l') = Inv_1(l) \wedge Inv_2(l')$ for all $(l, l') \in Loc_1 \times Loc_2$ and $((l_1, l_2), g, \sigma, p) \in prob$ if and only if one of the following conditions holds:

- $\sigma \in \Sigma_1 \setminus \Sigma_2$ and there exists $(l_1, g, \sigma, p1) \in prob_1$ such that $p = p_1 \otimes \eta_{(\emptyset, l_2)}$;
- $\sigma \in \Sigma_2 \setminus \Sigma_1$ and there exists $(l_2, g, \sigma, p2) \in prob_2$ such that $p = \eta_{(\emptyset, l_1)} \otimes p_2$;
- $\sigma \in \Sigma_1 \cap \Sigma_2$ and there exists $(l_1, g_1, \sigma, p1) \in prob_1$ and $(l_2, g_2, \sigma, p2) \in prob_2$ such that $g = g_1 \wedge g_2$ and $p = p_1 \otimes p_2$

where for any $l_1 \in Loc_1, l_2 \in Loc_2$:

$$p = p_1 \otimes p_2(X_1 \cup X_2, (l_1, l_2)) = p_1(X_1, l_1).p_2(X_2, l_2).$$

b) **Priced Timed Automata**

Priced timed automata [36] are an extension of timed automata whose clocks can evolve with various rates.

A Priced Timed Automaton (PTA) is a tuple A = (L,lo,X,Σ, E,R, I) where:

- L is a finite set of locations;
- lo $\in L$ is the initial location;
- X is a finite set of clocks;
- $L(X)$ is the set of guards over X;
- $U(X)$ is the set of invariants over X;
- $\Sigma = \Sigma_i \uplus \Sigma_o$ is a finite set of actions partitioned into inputs (Σ_i) and outputs (Σ_o);
- $E \subseteq L \times L(X) \times \Sigma \times 2^X \times L$ is a finite set of edges;
- $R : L \to \mathbb{N}^x$ assigns a rate vector to each location;
- $I : L \to U(X)$ assigns an invariant to each location.

A clock valuation over X is a mapping $\mathcal{V} : X \to \mathbb{R} \geq 0$, where $\mathbb{R} \geq 0$ is the set of non-negative reals. The valuation of a clock $x \in X$ after a time delay $d \in \mathbb{R} \geq 0$ is given by the formula $\mathcal{V}(x) + r(x) \times d$. Recalling that this valuation was defined by: $\mathcal{V}(x) + d$ in the case of ordinary timed automata.

A set of Priced Timed Automaton $A_j = (L_j, X_j, \Sigma_j, E_j, R_j, I_j)(j = 1..n)$ are composable into a closed network if:

- their clock sets are disjoint ($X_j \cap X_j = \emptyset$ when $j \neq k$);
- they have the same action set ($\Sigma = \Sigma_j = \Sigma_k$ for all j, k);
- their output action-sets provide a partition of $\Sigma(\Sigma_j o \cap \Sigma_k o = \emptyset$ for $j \neq k$).

Let $A_j = (L_j, X_j, \Sigma_j, E_j, R_j, I_j)$ with $(j = 1..n)$ composable NPTAs. Their composition $(A_1|...|A_n)$ is the NPTA $A = (L, X, \Sigma, E, R, I)$ where:

- $L = \times_j L_j$;
- $X = \cup_j X_j$;
- $R(l)(x) = R_j(l_j)(x)$ when $x \in X_j$;
- $I(l) = \cap_j I(l_j)$;
- $(l, \cap_j g_j, a, \cup_j r_j, \acute{l}) \in E$ whenever $(l_j, g_j, a, r_j, \acute{l_j}) \in E_j$ for $j = 1..n$.

Each Priced Timed Automaton decides based on a given delay density function and output probability function how much to delay before outputting and what output to broadcast at that moment. Obviously, in such a race between components the outcome will be determined by the component that has chosen to output after the minimum delay: the output is broadcast and all other components may consequently change state. The delay density function will be either a uniform in cases with time-bounded delays or an exponential distribution in cases of unbounded delays.

1.3.3 Properties Specification Formalisms

The principle of Model Checking is to verify that one or more properties are respected by a model of the system. These properties are [17]:

- **Reachability**: reachability property makes it possible to verify that a configuration is reachable by a series of transitions from another system configuration. Other types of properties are defined from Reachability.
- **Safety**: a safety property is a property stating that "something bad never happens". Generally, safety requirements include the absence of deadlocks and similar critical states that can cause the system to crash.

  Deadlock, sequential programs that are not subject to divergence (e.g., endless loops) have a terminal state, which has no outgoing transitions. For concurrent systems, however, computations typically do not terminate. In such case, terminal states are undesirable and mostly represent a design error. Apart from simple design errors where it has been forgotten to indicate certain activities, in most cases such terminal states indicate a deadlock. A deadlock occurs if the complete system is in a terminal state, although at least one component is in a (local) non-terminal state. A typical deadlock scenario occurs when components mutually wait for each other to progress.
- **Liveness**: different from safety properties, liveness properties mean that "something good" will eventually happen.

Properties are generally stated using temporal logics to specify how the system evolves over time. There are two main categories [26]:Linear Temporal Logic (LTL) and CTL(Computational Tree Logic). These two logics have derivatives to explicitly take into account the time and probabilities (PLTL, TLTL, PCTL, TCTL, PTCTL).

1.3.3.1 Linear Temporal Logic (LTL)

LTL [37] extends traditional propositional logic with temporal operators. Thus, LTL allows assertions about the temporal behaviour of a system where each moment in time has a unique future trajectory. An LTL formula ϕ has the following syntax:

$$\phi ::= p|(\neg\phi)|(\phi \wedge \phi)|(\phi U \phi)|(G\ \phi)|(F\ \phi)|(X\ \phi)$$

where:

- p is an atomic proposition,
- X_p: p holds next time,
- F_p: p holds sometime in the future,
- G_p: p holds globally in the future,
- ${}_pU_q$: p holds until q holds.

LTL semantics is originally defined by Pnueli [38] over infinite sequences of states that correspond to infinite or non-terminating sequences of computations. However, over the years, there has been more and more interest in run-time LTL verification that overcomes several inherent problems of model checking of full-scale models. Usually, LTL deals only with infinite behaviour. Finite traces could be tackled with certain workarounds, such as looping the last state of the finite trace.

1.3.3.2 Computation Tree Logic (CTL)

The branching time temporal logic, Computation Tree Logic(CTL) [39], is one of the most popular temporal logics in practice. CTL allows us to express a wide variety of branching time properties which can be verified in polynomial time (that is, the time complexity of CTL model checking is polynomial in the size of the state transition system times the length of the CTL formula). This makes CTL model checking computationally attractive as compared to the linear time temporal logic, LTL, and the more expressive branching time logic. CTL formulas can be recursively decomposed into sub-formulas which can be checked independently on the states of the system. Once this is done, the verification of the formula itself reduces to a simple question of reachability through the states marked by the sub-formulas. In this logic a path quantifier (A "for every path", E "there exists a path") can prefix an assertion composed of arbitrary combinations of the usual linear-time operators (X, F, G, U). More precisely, the syntax of CTL formulae is defined as follows:

$$\phi ::= p|\neg\phi|\phi \wedge \phi|\phi \vee \phi|\phi \rightarrow \phi|AX\ \phi|EX\ \phi|AF\ \phi|EF\ \phi|AG\ \phi|EG\ \phi|A[\phi U \phi]|E[\phi U \phi]$$

CTL formulae are interpreted over states in Kripke structures. Specifically, the CTL semantics is given by the operator $\models$ such that $K, s \models f$ means that the formula f is true in the state s of the Kripke structure K. All the CTL formulae are state formulae, but their semantics is defined using the intermediate concept of path formulae. In this context the notation $K, \pi \models f$ means that the formula f is true along the path π in the Kripke structure K. The operator π is defined inductively as follows:

1. $K, s \models \top$ is true and $K, s \models \bot$ false for any state s in any Kripke structure K.
2. $K, s \models a$, $a \in AP$(Atomic Propositions) if and only if $a \in L(s)$.($L(s)$ is the set of propositions that satisfied the state s).
3. $K, s \models \neg f$ if and only if : $\neg(K, s \models f)$ for any state formula f.
4. $K, s \models f \wedge g$ if and only if : $K, s \models f$ and $K, s \models g$ for any state formulae f and g.

5. $K, s \models f \vee g$ if and only if : $K, s \models f$ or $K, s \models g$ for any state formulae f and g.
6. $K, s \models Ef$ for some path formula f if and only if there exists a path $\pi = s \rightarrow s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_i, i \in \mathbb{N}$ such that $K, s \models f$.
7. $K, s \models Af$ for some path formula f if and only if $K, s \models Af$ for all paths $\pi = s \rightarrow s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_i, i \in \mathbb{N}$ such that $K, s \models f$.

We use π^i to denote the $i-th$ state of a path π, with the first state being π^0. The operator $\models$ for path formulae is then defined as follows:

1. $K, \pi \models Xf$ if and only if, $\pi^1 \models f$ for any state formula f.
2. $K, \pi \models fUg$ for any state formula f and g if and only if there exists $j \geq 0$ such that $K, \pi^k \models g$ for all $k \geq j$, $K, \pi^i \models f$ for all $i < j$.

1.3.3.3 Timed Computation Tree Logic (TCTL)

The branching-time logic TCTL [40] is a quantitative extension of CTL, where the scope of the temporal operators can be limited in time by subscripting them with time constraints .The syntax of TCTL is given by the following grammar:

$$\phi ::= p|\phi_1 \wedge \phi_2|\neg\phi|E(\phi_1 U^J \phi_2)|A(\phi_1 U^J \phi_2)$$

where J is an interval whose bounds are natural numbers.

The definition of the timed until is the essential part of TCTL. Given a path π and a timed automaton $TA, \pi \models \phi_1 U^J \phi_2$ if and only if:

1. $\exists i \geq 0. s_i + d \models \phi_2$ for some d such that:
$$\sum_{k=0}^{i-1} d_k + d \in J$$
2. $\forall 0 \leq j \leq i. s_j + d' \models \phi_1 \vee \phi_1$ for any d' such that:
$$\sum_{k=0}^{j-1} d_k + d' \leq \sum_{k=0}^{i-1} d_k + d$$

where for $s_i = (l_i, \nu_i)$ and $d \geq 0$ we have $s_i + d = (l_i, \nu_i + d)$.

The validity of "until" is defined by two conditions. The first one here states that property π must hold for some time in the interval J. Note that without specifying explicit clocks, interval J denotes a time interval counted from the start of the system. The first condition states that summing up the delay from this start time should be such that ϕ_2 holds for a time in J.

The second condition is more subtle. In particular, it requires that for all times before the time d at which ϕ_2 holds that either ϕ_1 or ϕ_2 holds. This is mainly due to the fact that ϕ_2 can hold too early but still be true in the time interval J.

The satisfaction relation is defined inductively as follows:

1. $s \models p$ if and only if: $a \in L(s)$,
2. $s \not\models p$ if and only if: $a \notin L(s)$,
3. $s \models \neg\phi$ if and only if: $s \not\models p$,
4. $s \models \phi_1 \wedge \phi_2$ if and only if : $s \models \phi_1$ and $s \models \phi_2$,
5. $s \models E\phi$ if and only if: $\exists \pi \in Paths_{div}(s). \pi \models E\phi$,
6. $s \models A\phi$ if and only if: $\forall \pi \in Paths_{div}(s). \pi \models E\phi$

1.3.3.4 Probabilistic Computation Tree Logic (PCTL)

The usual specification language for probabilistic model checking is Probabilistic Computation Tree Logic (PCTL) [41]. PCTL is an extension of the non-probabilistic Computation Tree Logic (CTL). PCTL provides a probabilistic operator whose role is to specify lower or upper probability bounds for reachability properties, in the sense of requiring that the probability of reaching a given set of states is above or below a given threshold value. The reachability properties can be constrained using the CTL path modality "until"U or its step-bounded variant $U^{\leq k}$. Using the probabilistic operator one might formally establish the guarantee that a system failure will occur within the next 100 steps with probability 10^{-8}or less, or that a leader will eventually be elected almost surely, that is, with probability 1. Besides the probability operator, expected cost operators can also be defined, which allow for reasoning, for example, about the average cost to reach a certain set of target states, or the accumulated cost within the next k steps. The cost operators can, for instance, be used to assert that the expected energy consumption within the next 100 steps is less than a given threshold.

The syntax of PCTL has two levels: one for the state formulae and one for the path formulae. The abstract syntax of state and path formulae is given respectively:

$$\Phi ::= \top | p | \Phi_1 \wedge \Phi_2 | \neg\Phi | \mathbb{P}_{\sim p}(\phi) | \mathbb{E}_{\sim c}(\diamond\Phi) | \mathbb{E}_{\sim c}(\leq K) | \mathbb{E}_{\sim c}(= K)$$

$$\phi ::= X\ \Phi | \Phi_1 U \Phi_2 | \Phi_1 U^{\sim c} \Phi_2$$

Where:

- $\top$ stands for the constant truth value "true",
- The operators $\mathbb{P}_{\sim p}(.)$ and $\mathbb{E}_{\sim c}(.)$ are called the probability and expectation operators,
- The subscripts $\sim p$ and $\sim c$ specify strict or non-strict lower or upper bounds for probabilities or costs,
- $\sim$ is a comparison operator $\leq, <, \geq$ or $>$,
- $p \in [0, 1]$ a rational threshold for probabilities,
- $c \in \mathbb{N}$ a non-negative integer that serves as a lower or upper bound for cumulated or instantaneous cost.

The PCTL state formula $\mathbb{P}_{\sim p}(\phi)$ asserts that, under all schedulers, the probability for the event expressed by the path formulaϕ meets the bound specified by $\sim p$.

Path formulas are built from one of the temporal modalities X(next) or U (until), where the arguments of the modalities are state formulas. No boolean connectors or nesting of temporal modalities are allowed in the syntax of path formulas. In addition to the standard until-operator, the above syntax for path formulas includes a cost-bounded version of until. The intuitive meaning of the path formula $\Phi_1 U^{\sim c} \Phi_2$ is that a Φ_2-state (i.e., some state where Φ_2 holds) will be reached from the current state along a finite path π that yields a witness of minimal length for the path formula $\Phi_1 U \Phi_2$ (i.e.,π ends in a Φ_2-state and all other states satisfy the formula $\Phi_1 \wedge \neg\Phi_2$) and where the total cost of π meets the constraint $\neg c$.

The expectation operator $\mathbb{E}_{\sim c}(.)$ enables the specification of lower or upper bounds for the expected cumulated or instantaneous cost. the state formula $\mathbb{E}_{\sim c}(\diamond\Phi)$ holds if the expected cumulated cost until a Φ-state is reached meets the requirement given by " $\sim c$" under all schedulers. Similarly, the state formula $\mathbb{E}_{\sim c}(\leq K)$ and $\mathbb{E}_{\sim c}(= K)$ assert the cost accumulated in the first k steps and the instantaneous cost at the k-th step, respectively, belong to the interval specified by " $\sim c$".

1.4 Conclusion

A sensor network is a collection of a large number of wireless sensing nodes that are spatially dispersed in a sensor field. Sensor nodes act as data generators and network relays, and they can sense, process data, and communicate with other sensor nodes. The end users of the data or administrators can then be able to make observations and respond to events in a particular environment. Wireless sensor nodes are very tiny and very cost effective. WSN is an up and coming technology that is being deployed for a myriad of applications at a very fast rate. However, if WSNs are to be widely adopted in fields where reliability and robustness are an issue, validation and verification methods of WSN systems must be effective.

In this chapter, we have described the main concepts related to wireless sensor networks such as: constraints, types, featured applications, and communication protocols. In addition, we gave an overview of the different methods, formalisms and tools available for formal verification and we focused on those that allow verifying protocols for WSNs.

In the next chapter, we will interest in MAC protocols in the context of wireless sensor networks.

Chapter 2

MAC Protocols for Wireless Sensor Networks

2.1 Introduction

The development of a reliable and energy-efficient protocol stack is important for supporting various WSN applications. Depending on the application, a network may consist of hundreds to thousands of nodes. Each sensor node uses the protocol stack to communicate with one another and to the sink. Hence, the protocol stack must be energy efficient in terms of communication and be able to work efficiently across multiple sensor nodes.

The energy source of a node is generally considered non-rechargeable. Thus, the most concern of the recent researches is placed in trade-offs between energy conserving and performance. A wireless radio is the most energy consuming unit of a node. It can operate in three or four different states: transmit, receive, idle and sleep [42]. However, all the active states consume almost the same energy. This problem can be addressed at software level, e.g. the network stack. A MAC layer is the most suitable level to address the energy inefficiency [43]. This layer is used to coordinate node access to the shared wireless medium. The MAC layer provides fine-grained control of the transceiver, and allows switching the wireless radio on and off. How frequent and when such switching have to be performed is the major goal of an energy saving mechanism of the MAC layer.

In this chapter, we focus on the MAC layer and especially on the challenges and constraints to design a MAC protocol. This chapter is organized as follows. Section 2.2 describes the communication models used in WSNs. Major sources of energy waste, power saving modes of operation, error detection techniques and the recovery techniques in wireless sensor network are introduced respectively in section 2.3. Section 2.4 presents the important attributes and metrics used to evaluate the performance of MAC protocols. Section 2.5 provides a survey on MAC Protocols for WSN including contention based, schedule based and hybrid protocols. A comparison between the different protocols is given in section 2.6. In section 2.7 various design protocols for cross-layer approach are highlighted. Section 2.8 describes the MAC protocols proposed for EH-WSNs. Finally, this chapter is concluded in section 2.9.

2.2 Communication Patterns

Different types of communication models are used to extract the behaviour of the sensor network traffic that has to be handled by a given MAC protocol. [44] defines three types of communication patterns in wireless sensor networks : broadcast, local gossip and converge-cast. A fourth type of communication pattern, multicast, is defined later in [45].

- **Broadcast**: is used by a base station to transmit some information to all sensor nodes of the network. Broadcasted information may include queries of sensor query-processing architectures, program updates for sensor nodes, control packets for the whole system. The broadcast type communication pattern should not be confused with broadcast type packet. For the former, all nodes of the network are intended receivers whereas for the latter the intended receivers are the nodes within the communication range of the transmitting node.

- **Local gossip**: in some scenarios, the sensors that detect an intruder communicate with each other locally. The sensors that detect the intruder, then, need to send what they perceive to the information center.

- **Converge-cast**: where a group of sensors communicate to a specific sensor. The destination node could be a cluster-head, data fusion center, base station.

- **Multicast**: in protocols that include clustering, cluster-heads communicate with their members and thus the intended receivers may not be all neighbours of the cluster-head, but just a subset of the neighbours.

2.3 MAC Protocol Design Challenges

The design of the MAC protocol in a WSN is subject to various constraints such as energy, topology, and network changes. We expose in the following sections the most popular strategies used to achieve energy efficiency and error control. But we start first with explanation of energy consumption sources.

2.3.1 Energy Consumption Sources

Minimizing energy to extend the network lifetime is a primary goal. The design of the MAC protocol should prevent energy wastage. Major sources of energy waste in wireless sensor network are basically of four types [46]:

- **Collision**: when a receiver node receives more than one packet at the same time, these packets are called "collided packets" even when they coincide partially. All packets that cause the collision have to be discarded and the re-transmissions of these packets are required which increase the energy consumption. Although some packets could be recovered by a capture effect, a number of requirements have to be achieved for its success.
- **Overhearing**: this occurs when a node picks up packets that are destined to other nodes.
- **Control packet overhead**: sending and receiving control Packets consume energy too and less useful data packets can be transmitted.
- **Idle listening**: listening to an idle channel to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, the sensor node will be in idle state for most of the time.

2.3.2 Power Saving Modes of Operation

Regardless of which type of medium access scheme is used for sensor networks, it certainly must support the operation of power saving modes for the sensor node. The most obvious means of power conservation is to turn the transceiver off when it is not required. Though this power saving method seemingly provides significant energy gains. In fact, if we blindly turn the radio off during each idling slot, over a period of time, we might end up expending more energy than if the radio had been left on. As a result, operation in a power saving mode is energy efficient only if the time spent in that mode is greater than a certain threshold. There can be a number of such useful modes of operation for the wireless sensor node, depending on the number of states of the micro-processor, memory, A/D converter and the transceiver. Each of these modes can be characterized by its power consumption and the latency overhead, which is the transition power to and from that mode. A dynamic power management scheme for wireless sensor networks is discussed in [47] where five power saving modes are proposed and inter-mode transition policies are investigated. The threshold time is found to depend on the transition times and the individual power consumption of the modes in question.

2.3.3 Error Detection

2.3.3.1 Error Detection Techniques

When transmitting a bit stream over a transmission line or channel, a scheme is normally incorporated into the transmission control circuit to enable the presence of bits or transmission error in the receiving block to be detected. In general, this is done by the transmitter which computes a set of additional bits based on the contents of blocks of bits to be transmitted. These extra bits are transmitted along with the original bits in the block. The receiver uses the complete sets of received bits to determine whether the block contains any error to the high probability. The two factors that determine the type of error detection scheme used are the bit error rate (BER) probability of the line and the type of error, that whether the errors occur as random single-bit errors or as groups of continuous of bit errors (burst error). The three most widely used schemes are parity, checksum and cyclic redundancy checks (CRC) [48][49].

a) **Parity checks**: in an **even** parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s is even. For **odd** parity schemes, the parity bit value is chosen such that there is an odd number of 1s. Receiver operation is also simple with a single parity bit. The receiver need only count the number of 1s in the received bits. If an odd number of 1-valued bits are found with an even parity scheme, the receiver knows that at least one bit error has occurred. More precisely, it knows that some odd number of bit errors have occurred. However, measurements have shown that, rather than occurring independently, errors are often clustered together in bursts .

b) **Checksumming methods**: in checksumming techniques, the d bits of data are treated as a sequence of k-bit integers. One simple checksumming method is to simply sum these k-bit integers and use the resulting sum as the error-detection bits. The 1s complement of this sum then forms the checksum that is carried in the segment header. The receiver checks the checksum by taking the $1s$ complement of the sum of the received data (including the checksum) and checking whether the result is all 1 bits. If any of the bits are 0, an error is indicated. Checksumming methods require relatively little packet overhead. For example, the checksums in TCP and UDP use only 16 bits. However, they provide relatively weak protection against errors.

c) **Cyclic Redundancy Check (CRC)**: an error-detection technique which is often used in the link layer is based on cyclic redundancy check codes. CRC codes operate as follows. Consider the d-bit piece of data, D, that the sending node wants to send to the receiving node. The sender and receiver must first agree on an $r + 1$ bit pattern, known as a generator, G. We will require that the most significant (leftmost) bit of G be a 1. The key idea behind CRC codes is: for a given piece of data, D, the sender will choose r additional bits, R, and append them to D such that the resulting $d + r$ bit pattern (interpreted as a binary number) is exactly divisible by G (i.e., has no remainder) using modulo-2 arithmetic. The process of error checking with CRCs is thus simple: The receiver divides the $d + r$ received bits by G. If the remainder is non-zero, the receiver knows that an error has occurred; otherwise the data is accepted as being correct.

2.3.3.2 Recovery Techniques

If there are error blocks, the receiver will initiate the recovery process to retrieve those error blocks after receiving a certain number of frames. Recovery techniques in WSN include

Automatic Repeat reQuest (ARQ) [50], Forward Error Correction (FEC), Hybrid ARQ (HARQ) [51], Simple Packet Combining (SPaC)[52], and Multi-Radio Diversity (MRD) [53]. We will expose hereafter the most important techniques: ARQ, FEC and HARC.

a **Automatic repeat request**: ARQ uses acknowledgement and time-out to provide explicit feedback to the sender. The feedback can be in the form of a positive acknowledgement (ACK) or a negative acknowledgement (NACK). The sender receiving a NACK or timing out will retransmit the data frame. A limitation to ARQ is that it is limited to frame error detection. An entire frame has to be retransmitted if there is a single bit error.The application of ARQ schemes is thus far unexplored in the regime of sensor networks. The usefulness of ARQ in sensor network applications is limited by the additional retransmission cost and overhead.

b **Forward error correction**: the idea of FEC is to get the transmission right the first time. For this purpose, FEC transmits together with original data some redundant data, called parities, to allow reconstruction of lost packets at the receiver. The redundant data is derived from the original data using techniques from coding theory: using the exclusive OR (XOR) operation allows one parity packet to be computed for a given set of original packets; using Reed-Solomon codes, multiple independent parities can be computed for the same set of packets. Reed-Solomon codes allow to achieve optimal loss protection, but lead to higher processing costs than schemes based on XOR operations. It is possible to compute several parity packets with lower processing cost than Reed-Solomon codes, but also with sub-optimal loss protection, by arranging original packets in a matrix and computing XOR parities over rows, columns or diagonals, or by constructing new codes based on XOR operations. FEC schemes do not need a return path. The recovery of lost data by reconstruction at the receiver requires very little time, which makes FEC attractive for applications with real-time requirements.

c **Hybrid ARQ (HARQ)**: a major difficulty when using FEC is to choose the right amount of redundancy in face of changing network conditions. Also, sending redundant data consumes additional bandwidth. In order to overcome this problem, ARQ and FEC can be used in combination:

- The first approach that combines ARQ and FEC, referred to as **hybrid ARQ type I**, immediately sends a certain amount of redundant data using FEC. If the loss rate obtained after reconstruction at the receiver is still too high, ARQ is used to retransmit. Using this approach it is possible to assure with a high probability that a large number of receivers obtain the data without retransmissions.
- Another possibility for combining ARQ and FEC, referred to as **hybrid ARQ type II**, is not to send any redundant data with the first transmission, but to send parity data when a retransmission is required. This approach is very bandwidth-efficient for reliable multicast to a large number of receivers. Error recovery by multicast retransmission of the original data packets requires retransmission of all lost packets. On the other hand, retransmission of a single parity packet allows all receivers to recover their lost packet. For a growing number of receivers and uncorrelated loss, the mean number of losses a single parity packet can repair is also growing.

2.4 Performance Metrics for MAC Protocols

To design a good MAC protocol for the wireless sensor networks, various attributes must be considered [46]. The first is energy efficiency. Sensor nodes are likely to be battery powered, and it is often very difficult to change or recharge batteries for these nodes. Prolonging their network lifetime is a critical issue. The second is robustness and scalability to the change in network size, node density and topology. Some nodes may die over time; some new nodes may join later; some nodes may move to different locations. The network topology changes over time as well due to many reasons. A good MAC protocol should easily accommodate such network changes. Other typical performance figures like fairness, throughput, or delay tend to play a minor role in sensor networks. Fairness is not important since the nodes in a WSN do not represent individuals competing for bandwidth, but they collaborate to achieve a common goal.

In order to evaluate and compare the performance of conscious MAC protocols, the following matrices are being used by the research community [54].

- **Energy consumption per bit**: the energy efficiency of the sensor nodes can be defined as the total energy consumed/total bits transmitted.
- **Average delivery ratio**: the average packet delivery ratio is the number of packets received to the number of packets sent averaged over all the nodes.
- **Average packet latency**: the average packet latency is the average time taken by the packets to reach to the sink node.
- **Network throughput**: the network throughput is defined as the total number of packets delivered at the sink node per time unit.

2.5 Classification of MAC Protocols for WSN

Several MAC protocols have been successfully proposed to meet the stringent design requirements of WSNs. These protocols depend on how protocol allows nodes to access the channel. They can be classified broadly into three categories: contention-based, schedule-based and hybrid protocols.

2.5.1 Contention-based Protocols

In contention schemes a common channel is shared by all nodes and it is allocated on demand. At any moment, a contention mechanism (usually based on back-off schemes) is employed to decide which node has the right to access the channel. Because contention protocols allocate resources on demand, they can scale more easily across changes in node density or traffic load and can be more flexible as topologies change. There is no requirement to form communication clusters, and peer-to-peer communication is directly supported. Moreover, contention protocols do not require fine-grained time synchronization. The major disadvantage of a contention protocol is its inefficient usage of energy [55].

The representative contention-based protocols are:

a **Additive Link On-ine HAwaii system(ALOHA)**: a node simply transmits a packet when it is generated (pure ALOHA) or at the next available slot (slotted ALOHA). Should the transmission be unsuccessful, every colliding user, independently of the others, schedules

its retransmission to a random time in the future. This randomness is required to ensure that the same set of packets does not continue to collide indefinitely.

b **Carrier Sense Multiple Access(CSMA)**: the main idea is listening before transmitting. The purpose of listening is to detect if the medium is busy, also known as carrier sense. The channel is sensed and if found idle the packet is transmitted. When a collision takes place, each node waits for a random time before accessing the channel again. In accordance with common networking lore, CSMA methods have a lower delay and promising throughput potential at lower traffic loads, which generally happens to be the case in WSNs. CSMA is popular because it is simple, flexible and durable. It does not need much infrastructure support. It does not require clock synchronization and global topology knowledge. When a node joins or leaves the network dynamically it can be controlled without an extra operation. After all, a node can receive a packet from two different nodes which are not in the same coverage area. Packet collision occurs thusly. This problem is known in literature as a hidden terminal problem. This problem leads to energy loss in sensor applications. Fortunately, hidden terminal problems can be alleviated by using a RTS/CTS operation. However, additional load comes to the network due to RTS/CTS messages because data packets are small in the sensor networks.

Several approaches for MAC that utilize collision avoidance and reduce both contention and idle listening have been proposed to improve energy efficiency. Most of theme are CSMA-based. In the following sections, we review some of the representative approaches.

2.5.1.1 IEEE 802.11

In the IEEE 802.11 standard [56], the medium access mechanism, called the Distributed Coordination Function, is basically a Carrier Sense Multiple Access with Collision Avoidance mechanism (CSMA/CA). In such a manner that, before a packet transmission, the nodes have to listen to the transmission channel to determine whether other nodes are transmitting. If the medium is sensed as free for a specified amount of time, the node is allowed to begin its transmission. However, if the medium is sensed as busy, the node defers its transmission for a random period of time, called a back-off period. The receiving node checks the CRC of the received packet and sends an acknowledge packet (ACK) after waiting for a specified amount of time once the packet is received. If an ACK is not received, the packet is considered lost and a retransmission is arranged. In order to reduce the probability of two stations colliding due to not hearing each other, which is well known as the "hidden node problem", the standard defines a Virtual Carrier Sense mechanism: a station wanting to transmit a packet first transmits a short control packet called RTS (Request To Send), which includes the source, destination, and the duration of the intended packet and ACK transaction. The destination station responds (if the medium is free) with a response control packet called CTS (Clear to Send), which includes the same duration information. The performances of CSMA/CA are strictly related to the network topology and the nodes density. Inevitably, large latency times affect the efficiency of the system, because before transmitting each station has to wait an unpredictable amount of time that mainly depends on the demands of users and topology of the network.

2.5.1.2 B-MAC

The Berkeley Media Access Control (B-MAC) [57] is designed based on CSMA mechanism and especially for low power WSNs. In order to reduce energy consumption, B-MAC provides Clear Channel Assessment (CCA) and packet back off, link-layer acknowledgement,

and Low Power Listening (LPL). For collision avoidance, B-MAC utilizes CCA to determine if the channel is clear. CCA is an outlier algorithm that searches for outliers in the received sample signals. An outlier exists if the channel energy is significantly below the noise floor. During the channel sampling period, if an outlier is found, the channel is clear, else the channel is busy. In case of a busy channel, packet back-off is used. Back-off time is either initially defined or randomly chosen.

B-MAC supports link-layer acknowledgement for unicast packets. When the receiver receives a packet, an acknowledgement packet is sent to the sender. To reduce power consumption, B-MAC employs an adaptive preamble sampling scheme called LPL. LPL performs periodic channel sampling by cycling through awake and sleep periods. In the awake period, the node's radio is turned on to check for activities in the channel using CCA. If activities are detected, it will remain awake to receive the incoming packet. Once it receives the packet, it will go back to sleep. Idle listening occurs when the node is awake but there is no activity in the channel. A time-out will force the node to go back to sleep.

2.5.1.3 S-MAC

The Sensor-MAC (S-MAC) [58] is a CSMA based protocol which reduces idle listening by periodically putting nodes into sleep state. In the sleep state, the radio is completely turned off. In S-MAC, the low-duty-cycle mode is the default operation of all nodes. They only become more active when there is traffic in the network. To reduce control overhead and latency, S-MAC introduces coordinated sleeping among neighbouring nodes. Neighbouring nodes within a virtual clusters follow the same sleep/listen schedule and the neighbouring nodes in two different virtual clusters follow the periods of both clusters.

Schedule exchanges are accomplished by periodical SYNC packet broadcasts to immediate neighbours. The period for each node to send a SYNC packet is called the synchronization period. Figure 2.1 represents a sample sender-receiver communication. Collision avoidance is achieved by a carrier sense. Furthermore, RTS/CTS packet exchanges are used for unicast type data packets.

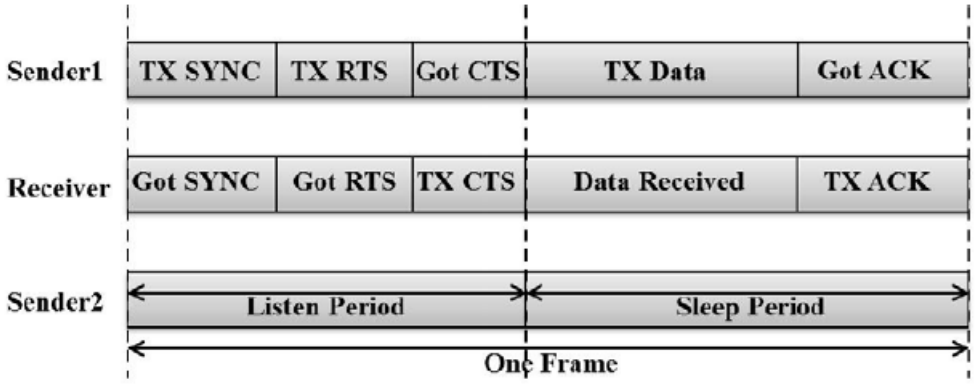


Figure 2.1: S-MAC operation mechanism

2.5.1.4 T-MAC

T-MAC (Timeout MAC) [59] is proposed as an enhanced version of S-MAC. Since the parameters of S-MAC (such as listen and sleep periods) are constant and cannot be changed after deployment, it is not suitable for variable traffic load, which is common in WSNs. In addition, in

networks without any traffic, S-MAC still has to wake up periodically and this causes significant energy waste. Hence, the design of T-MAC is to offer a dynamic and configurable duty cycle by modifying the listen-sleep period to improve the S-MAC's poor performance on energy consumption and variable traffic support.

T-MAC replace the fixed active time with adaptive active time. The adaptation is based on a monitoring of the activation events (like data reception and transmission) of a node. If no activation event has appeared after the specified time, the node goes to sleep. Therefore, all the traffic must be buffered between activity periods and sent in bursts at the beginning of the next active period. The advantage of T-MAC is that very low duty cycles can be obtained, but at the expense of high latency and a collapse under high loads.

2.5.2 Schedule-based Protocols

In schedule based protocols, each node is given a guaranteed periodic access to the shared medium by segmenting the channel into super-frames and a global synchronization between nodes is assumed. A slot is reserved to each node and the node uses the same slot in subsequent super-frames. Since slots are pre-allocated to individual nodes, they are collision-free. These protocols are characterized by a duty cycle built-in with the inherent collision-free nature that ensures low energy consumption. On the other side, the complexity of the design is high due to problems of synchronization. In general, they are not flexible to changes in node density or movement, and lack of peer-to-peer communication [46].

The representative schedule-based protocols are:

a **Time Division Multiple Access (TDMA)**: in TDMA based protocols, packet collision, unintentional receiving and unnecessary listening to the medium can be avoided by utilizing sending and listening periods, but a strict synchronization is needed. It allows nodes to share the same frequency channel by dividing the signal into different time-slots. It supports low duty cycle operation: a node only needs to turn on its radio during the slot that it is assigned to transmit or receive. On the other hand, TDMA provides a solution to the hidden terminal problem without a need for extra messages because it programs the transmission time of neighbour nodes at different times. However, deterministic TDMA scheduling requires a large overhead in order to maintain accurate synchronization between sensors and to exchange local information, such as the network topology and the communication pattern. Furthermore, the latency increases linearly with the total number of sensors sharing the channel since TDMA assigns a separate time-slot to each transmitting sensor [60].

b **Frequency Division Multiple Accesses (FDMA)**: it allocates users with different carrier frequencies of the radio spectrum. It is another scheme that offers a collision-free medium, but it requires additional hardware to dynamically communicate with different radio channels. This increases the cost of the sensor nodes, which is in contrast with the philosophy of sensor network systems.

c **Code Division Multiple Access (CDMA)**: it employs spread spectrum technology and a special coding scheme (where each transmitter is assigned a code) to allow multiple users to be multiplexed over the same physical channel. It also offers a collision-free medium, but its high computational requirement is a major obstacle for the minimum energy consumption objective in WSNs.

Several protocols have been proposed under this category. In the following sections, we review some of TDMA-based MAC protocols.

2.5.2.1 TRAMA

TRaffic Adaptive Medium Access (TRAMA) [61] is a TDMA based protocol. TRAMA assumes a single, time-slotted channel for both data and signalling transmissions. Figure 2.2 shows the overall time-slot organization of the protocol. Time is organized as sections of random-access (signalling slots) and scheduled-access periods (transmission slots).

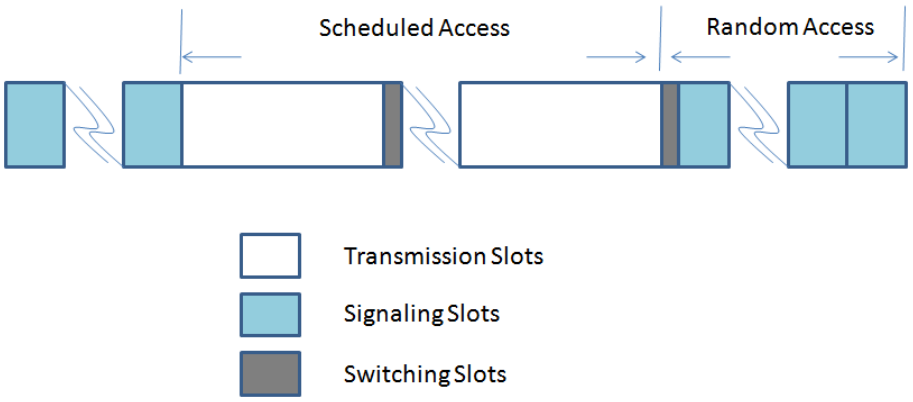


Figure 2.2: Time slot organization.

TRAMA consists of three components: the Neighbour Protocol (NP), the Schedule Exchange Protocol (SEP) and the Adaptive Election Algorithm (AEA).

- **Neighbour protocol**: NP propagates one-hop neighbour information among neighbouring nodes during the random access period using the signalling slots to obtain consistent two-hop topology information across all nodes. During the random access period, nodes perform contention-based channel acquisition and thus signalling packets are prone to collisions.
- **Schedule exchange protocol**: nodes use SEP to exchange schedules with neighbours. Essentially, schedules contain current information on traffic coming from a node, i.e., the set of receivers for the traffic originating at the node. A node has to announce its schedule using SEP before starting actual transmissions.
- **Adaptive election algorithm**: AEA selects transmitters and receivers to achieve collision-free transmission using the information obtained from NP and SEP

Transmission slots are used for collision-free data exchange and also for schedule propagation.

2.5.2.2 D-MAC

The main idea behind DMAC [62] is that the data delivery paths from sources to sink are in a tree structure, a data gathering tree. Flows in the data gathering tree are unidirectional from sensor nodes to sink. There is only one destination, the sink. All nodes except the sink will forward any packets they receive to the next hop. The activity schedule of nodes is staggered on the multi hop path to wake up sequentially. DMAC is proposed to deliver data along the data gathering tree, aiming at both energy efficiency and low latency.

Figure 2.3 shows a data gathering tree and the staggered wake-up scheme. An interval is divided into receiving, sending and sleep periods. In receiving state, a node is expected to receive a packet and send an ACK packet back to the sender. In the sending state, a node will

try to send a packet to its next hop and receive an ACK packet. In sleep state, nodes will turn off radio to save energy. The receiving and sending periods have the same length which is enough for one packet transmission and reception.

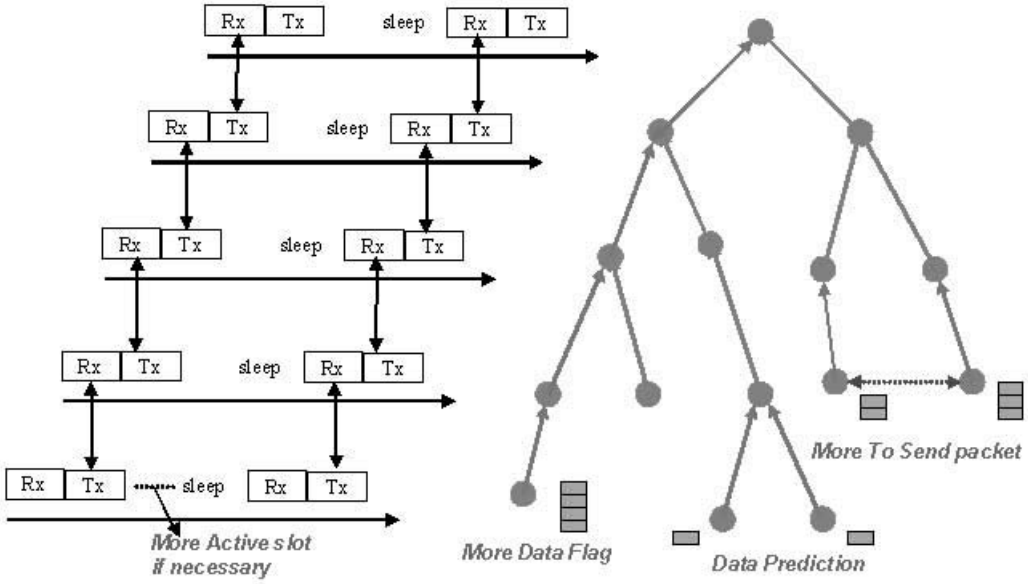


Figure 2.3: DMAC in a data gathering tree.

2.5.2.3 DEE-MAC

DEE-MAC (Dynamic Energy Efficient) [63] is a cluster and TDMA based protocol that reduces energy consumption by using cluster head assigned slots to keep synchronization on the data transmission/reception schedule and force other idle nodes to sleep mode. Each cluster is dynamically formed based on the remaining power as all nodes contend to be the cluster head. Process of joining and leaving the cluster head is performed freely. DEE-MAC operations comprise of rounds. Each of the rounds includes a cluster formation phase and a transmission phase. In the cluster formation phase, a node decides whether to become the cluster head based on its remaining power. The node with the highest power level is elected as the cluster head. Each new round introduces formation of another cluster with different group of nodes based on the current node power level and the network structure changes. After the successful cluster head election, the system enters the transmission phase. This phase comprises of a number of sessions and each of the session consists a contention period and a data transmission period. For the time of the contention period, each of the nodes keeps their radio on, and indicates interest to send a packet to the cluster head.

After this period, the cluster head knows which of the node has data to transmit. The cluster head builds a TDMA schedule that is broadcasted to all nodes. Each of the nodes is assigned with one data slot in each session. Based on the broadcasted schedule each of the nodes, having a data to receive or send, is awaken.

2.5.3 Hybrid Protocols

The basic idea of hybrid-based MAC protocols is to achieve better energy performance by combining the advantages of contention-based and schedule-based elements. Hybrid protocol divides the channel into two parts, channel control packets and data packets, in channel control

packets data is sent in the random access and in data packets data are transmitted in the scheduled channel. The hybrid protocols can save higher energy and supply better scalability and flexibility in comparison to these two methods.

2.5.3.1 IEEE 802.15.4

The international standard IEEE 802.15.4 [64] specifies the MAC sub-layer and the Physical Layer for LR-WPAN (hereafter denoted as PAN). The IEEE 802.15.4 protocol is very much associated with the ZigBee protocol [65] which specifies the protocol layers above IEEE 802.15.4 to provide a full protocol stack for low-cost, low-power, low data rate wireless communications. This protocol provides enough flexibility for fitting different requirements of WSN applications by adequately tuning its parameters, even though it was not specifically designed for WSNs. In fact, low-rate, low-power consumption and low-cost wireless networking are the key features of the IEEE 802.15.4 protocol, which typically fit the requirements of WSNs.

The IEEE 802.15.4 standard defines three types of device:

- **PAN coordinator**: is the core controller of PAN, and is responsible for the operation and management of the whole network.
- **Coordinators**: collaborate with each other as well as with the PAN coordinator to maintain the connection and functionality of a subset of nodes in the network.
- **Nodes**: can only communicate with (PAN) coordinators for data exchange without any network organization functions (e.g., routing).

With the above three devices, the IEEE 802.15.4 based network can be formed into a star topology, a cluster-tree topology as well as a mesh topology.

Two channel access methods are supported by IEEE 802.15.4 MAC layer, which are beacon enabled mode and non-beacon enabled mode.

In beacon enabled mode, a special duty cycle is scheduled by means of a super-frame structure which is bounded by beacons. This super-frame structure, shown in Figure 2.4, is periodically generated by the coordinator and broadcast to all nodes for synchronization purposes. Each super-frame is composed of an active period and an inactive period. Communication between nodes and coordinators occurs during the active period. The active period can be further divided into a contention access period (CAP) and a contention free period (CFP). During CAP, nodes adopt a slotted CSMA/CA algorithm for channel access. During CFP, a certain number of guaranteed time slots (GTS) can be assigned to specific nodes so that the communication between nodes and coordinator cannot be interfered by any collisions in these GTS. During the inactive period, nodes enter into a low power sleep state for energy saving and wake up at the beginning of the next super-frame.

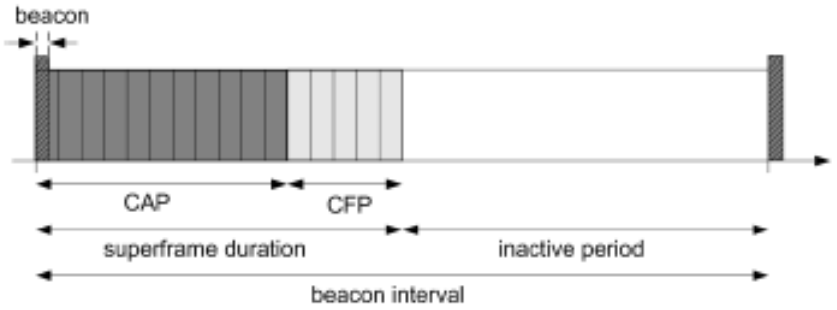


Figure 2.4: Super-frame structure

For the non-beacon enabled mode, this is totally contention-based, since the nodes are always in the active state and use an unslotted CSMA/CA algorithm for channel competition and data communication. Compared with the non-beacon enabled mode, the beacon enabled mode is more energy-efficient due to the low power consumption in the sleep schedule but at the cost of network latency and scalability. Thus, the trade-off should be carefully selected based on specific application requirements.

2.5.3.2 WISE-MAC

In WISE-MAC [66] all sensor nodes are defined to have two communication channels. Data channel is accessed with TDMA method, whereas the control channel is accessed with CSMA method. Wise MAC protocol uses non-persistent CSMA (np-CSMA) with preamble sampling to decrease idle listening. In the preamble sampling technique, a preamble precedes each data packet for alerting the receiving node. All nodes in a network sample the medium with a common period, but their relative schedule offsets are independent.

If a node finds the medium busy after it wakes up and samples the medium, it continues to listen until it receives a data packet or the medium becomes idle again.The size of the preamble is initially set to be equal to the sampling period. However, the receiver may not be ready at the end of the preamble, due to reasons like interference, which causes the possibility of over emitting type energy waste. To reduce the power consumption incurred by the predetermined fixed-length preamble, Wise MAC offers a method to dynamically determine the length of the preamble. That method uses the knowledge of the sleep schedules of the transmitter node's direct neighbours. In that way, every node keeps a table of sleep schedules of its neighbours. Based on neighbours' sleep schedule table, Wise MAC schedules transmissions so that the destination node's sampling time corresponds to the middle of the sender's preamble.

2.5.3.3 Z-MAC

Z-MAC (Zebra-MAC) [67] is a hybrid MAC protocol that combines the strength of the TDMA and CSMA while offsetting their weaknesses. The main feature of Z-MAC is its adaptability to the level of contention in the network under low contention, it behaves like CSMA, and under high contention, like TDMA. In Z-MAC, a time slot assignment is performed at the time of deployment. After the slot assignment, each node reuses its assigned slot periodically in every predetermined period, called frame. Each node is owner of one or more slots, a node may transmit during any time slot. Before a node transmits during a slot (not necessarily at the beginning of the slot), it always performs carrier-sensing and transmits a packet when the channel

is clear. However, an owner of that slot always has higher priority over its non-owners in accessing the channel. The priority is implemented by adjusting the initial contention window size in such a way that the owners are always given earlier chances to transmit than non-owners.The goal is that, during the slots where owners have data to transmit, Z-MAC reduces the chance of collision since owners are given earlier chances to transmit and their slots are scheduled a priori to avoid collision, but when a slot is not in use by its owners, non-owners can steal the slot. This priority scheme has an effect of implicitly switching between CSMA and TDMA depending on the level of contention. An important feature of this priority scheme is that the probability of owners accessing the channel can be adjusted independently from that of non-owners.

Z-MAC has a set-up phase in which it runs the following operations in sequence:neighbour discovery,slot assignment,local frame exchange, and global time synchronization.

- **Neighbour discovery and slot assignment**: as a node starts up, it first runs a simple neighbour discovery protocol where it periodically broadcasts a ping to its one-hop neighbours to gather its one-hop neighbour list. Through this process, each node gathers the information received from the pings from its one-hop neighbours, which essentially constitutes its two-hop neighbour information. The two-hop neighbour list is used as input to a time-slot assignment algorithm. This assignment guarantees that no transmission by a node to any of its one-hop neighbours interferes with any transmission by its two-hop neighbours.

- **Local frame exchange**: once a node picks a time slot, each node needs to decide on the period in which it can use the time slot for transmission. This period is called the time frame of the node. Each node maintains its own local time frame that fits its local neighbourhood size, but avoids any conflict with its contending neighbour. Every node forwards its frame size and slot number to its two-hop neighbourhood. At this point, nodes are finally ready to run the transmission control. A node can be in one of two modes:low contention level(LCL) or high contention level(HCL). A node is in HCL only when it receives an explicit contention notification message from a two-hop neighbour, otherwise, the node is in LCL. In LCL, any node can compete to transmit in any slot, but in HCL, only the owners of the current slot and their one-hop neighbours are allowed to compete for the channel access.

- **Global time synchronization**: Z-MAC requires clock synchronization under high contention. However, note that synchronization is required only among neighbouring senders and when they are under high contention. This offers an excellent opportunity to optimize the overhead of clock synchronization because synchronization is required only locally among neighbouring senders, and the frequency of synchronization can be adjusted according to the transmission rates of senders so that senders with higher data rates transmit more frequent synchronization messages. In this scheme, receivers passively synchronize their clocks to the senders'clocks and do not have to send any synchronization messages.

2.6 Comparison of MAC Protocols

In summary, contention-based protocols are much widely studied in WSN and generally based on or similar to CSMA. When one node has a packet to send, it will have to struggle with the other competitors to get permitted of using the medium. The winner selection is somehow randomized. The synchronous MAC protocols are generally duty-cycled and require time-synchronized, such as S-MAC and T-MAC. The state-of-the-art synchronization method is to

be done through hardware or message exchange, and then a piggybacked acknowledge can be used to solve the clock shifting effect. The asynchronous versions use LPL or its preamble-shortened approach to match up the transmission period between transmitter and receiver end, such as B-MAC and WISE-MAC.

Among them, [68] compared the power dissipation between asynchronous and synchronous contention-based protocol. In which, LPL method is interesting in very low traffic intensity (less than one packet per day)or dynamically changing topologies. Otherwise, synchronous protocol outperforms asynchronous one. The drawback of such a protocol is the packet collision causing by increasing network density and hidden terminal.

In comparison to contention-based protocols, Schedule-based protocols are generally centralized and suitable for static topologies. Assigned nodes play the master role to allocate slotted network resource to their slaves. The mechanism is generally based on TDMA or CDMA. The clock of each node must be time-synchronized. Scheduled slots can be fixed or on demand. These protocols use TDMA as the baseline MAC scheme, and then take CSMA, Aloha or CDMA for improving these join/leave/synchronize messages. The drawbacks are firstly not easy to adapt to the dynamics of network, and secondly the slower response of the centralized control while adapting the schedule to the traffic variation.

To combine both strengths of schedule and contention based protocols, hybrid protocols are very helpful in finding a trade-off between energy and other performance. The disadvantage is that it could sometimes be too complex to be applied for a quick and large scale of deployment.

2.7 Cross-layer Approaches

While with the extending range of WSN application, performance of the WSN has to be improved to satisfied increasing requirements from variety of customers. In this condition, cross-layer design for WSN has been considered. Though the network communication can be achieved conveniently by employing layered structure, the side effect such QoS concession, latency, additional overload, caused by blocking between each layer cannot be ignore [69].

The cross-layer design is an emerging technique for both wireless and wired networks. The theme behind this approach is optimizing the flow of data in such a way that any two or more layers can be crossed for upgrading the complete performance of the network [70]. The ultimate goal is to enhance the performance of WSNs in terms of energy usage, routing, delivery of multimedia contents, and network management. Depending on the user requirements, two, three, or four layers of the layered architecture can be crossed. For example, in order to accomplish in network processing and better multi-path selection, the application layer and routing layer are combined. Service level differentiation, priority scheduling, and efficient routing techniques are achieved when the routing and MAC layers work together. Moreover, achieving QoS and reliability of multimedia content when multiple routing paths scheme is used, the routing and transport layer contribute in achieving these enhancements [71]. The cross-layer design approach is one of the preposition in this regard that enables the designers to develop protocols which enable communications across layers in contrast to the layered architecture approach [72].

Various design solutions are proposed to explore the benefits of a cross-layer approach. Below, we will present these proposals.

2.7.1 MAC-CROSS protocol

The design goal of MAC-CROSS [73] is to minimize energy consumption by continuously turning off the radio interface of unnecessary nodes that are not included in the routing path. So, it combines the functionalities of the network and the medium access protocols.

In this protocol, nodes are classified into three types depending upon the state defined by data transmission: Communicating Parties, Upcoming communicating Parties and Third Parties. A state may dynamically change whenever data traffic is transmitted.

- **Communicating Parties (CP)**: any node currently participating in the actual data transmission.
- **Upcoming communicating Parties (UP)**: any node to be involved in the actual data transmission.
- **Third Parties (TP)**: any nodes that are not included on a routing pathand hence not involved in the actual data transmission at all.

In MAC-CROSS, only a few nodes concerned of the actual data transmission(i.e., the necessary UP nodes) are asked to wake up, while other TP nodes can continuously remain in their sleep modes.

The format of RTS/CTS control frames are modified from their originals in S-MAC protocol family. This modification is for informing a node the fact that its state is changed to UP or TP in the corresponding listen/sleep period. Figure 2.5.a shows the original RTS and CTS control packet formats, Figure 2.5.b and c show the new RTS and CTS control packet formats for MAC-CROSS.

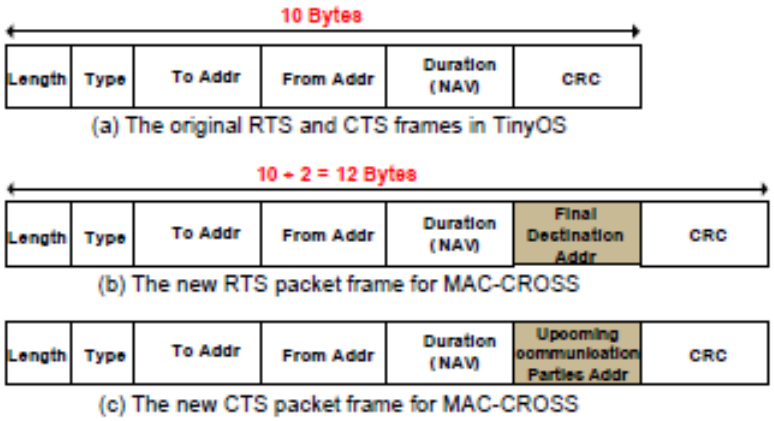


Figure 2.5: RTS and CTS frames in MAC-CROSS

The new RTS and CTS packet add only one field to the original packets. The newly added field in RTS is Final_Destination_Addr, by which the receiver's routing agent can search for the next hop address. The new field of CTS is UP_Addr and it informs which node is UP to its neighbours.

When a node receives an RTS packet including the final destination address of sink, its routing agent refers to the rouging table for getting the UP and informs back to its own MAC. The MAC agent then transmits CTS packet including the UP information. After receiving the CTS packet , the node changes its state to UP and other neighbour nodes become aware of the

fact that they are TP nodes. UP node has to wake up when NAV timer expires for receiving data, but other nodes continuously sleep even if NAV timer expires for saving energy. Otherwise, if no such information about UP is available in node's routing agent, it means the routing path is broken or has not yet been established. In this case, MAC-CROSS is performed just like S-MAC without cross layer concept.

2.7.2 XLM Protocol

XLM protocol [74] combines the functionalities of the transport, network and medium access protocols into a single module. XLM is built upon the concept of letting a node decide whether it wants to participate in communication or not.

A node initiates transmission by broadcasting an RTS packet to indicate its neighbours that it has a packet to send. Upon receiving an RTS packet, each neighbour of node decides to participate in the communication or not. This decision is given through initiative determination. The initiative determination is a binary operation based on a set of four conditions. All four conditions must be satisfied for a node to participate. The first condition ensures reliable links be constructed for communication. For this purpose, it requires that the received signal to noise ratio (SNR) of an RTS packet, is above some threshold for a node to participate in communication. The second condition prevents congestion by limiting the traffic a node can relay. The third condition ensures that the node does not experience any buffer overflow and hence, also prevents congestion. The last condition ensures that the remaining energy of a node stays above a minimum value. This constraint guarantees even distribution of energy consumption.

Using the initiative concept, XLM performs local congestion control, hop-by-hop reliability, and distributed operation.

2.7.3 EYES MAC Protocol

EYES MAC protocol [75] exploits the benefits of cross-layer interaction between the network and data-link layers. The approach addresses a self-organizing medium access control(MAC) protocol that uses an algorithm to decide the grade of participation of a sensor node in creating a connected network based on local information only, and a tightly integrated, efficient routing protocol.

In EYES MAC protocol three modes of operation in: active, passive and dormant. When a node is in active mode, it will contribute to the network by taking part in forwarding messages to a destination and accepting data from passive nodes. Passive nodes, on the other hand, conserve energy by only keeping track of active nodes, which can forward their data and inform them of network-wide messages. The nodes in dormant mode put themselves in a low-power state for an agreed amount of time or, for example, when their power source runs out of energy and has to be charged again using ambient energy like light.

The medium access protocol is based on TDMA. But unlike traditional TDMA-based systems, the time slots are not divided among the networking nodes by a central manager. A time slot is divided into three sections: Communication Request (CR), Traffic Control (TC), and the data section. In the CR section other nodes can make requests to the node controlling the current time slot. Nodes that have a request to the time slot owner will pick a random start time in the short CR section to make their request. Communication in this section is not guaranteed to be collision-free. Nodes that do not have a request for the current slot owner will keep their transceiver in a low-power state during the entire CR section.

The owner of a time slot will always transmit a TC message in the time slot, regardless of whether or not a request was filed. All nodes within one-hop distance of the controller of the current timeslot will put effort into receiving this message, since this message is used for synchronization purposes and control information. When a time slot is not controlled by any node, all nodes remain in sleep state during that time slot. The time slot owner also indicates in its TC message what communication will take place in the data section. By listening to TC sections of neighbouring nodes, nodes have knowledge of local topology.

The EYES Source Routing (ESR) algorithm is an on-demand algorithm that enables dynamic,self starting, multi-hop routing to be established when a source sensor node wishes to send a data packet. The ESR algorithm has three phases: route set-up, route maintenance, and route re-establishment. The routing protocol is essentially applied on the connected active set only, which implies that passive nodes should forward data to one of the nodes of this set first. The protocol utilizes the topology information already provided by the MAC protocol to efficiently manage topology changes due to mobility, node and communication failures, and power duty cycling.

2.8 Energy Harvesting MAC Protocols

Energy harvesting technologies [76] can prolong WSN lifetime by converting solar, wind, vibrational, thermal energy into electrical energy. Their disruptive potential has led to the formulation of the so called Energy Harvesting-Wireless Sensor Networks (EH-WSNs). The effectiveness of EH-WSNs mainly depends on the interplay between EH technologies and the protocol stack [77].

With EH-WSNs, MAC design becomes even more challenging because the pattern of energy harvested from the environment is not easily predictable in advance. Although, it can be predicted up to short or medium time intervals that can be of the order of microseconds to hours depending on various factors including but not limited to application, topology, energy harvesting technique and, the environment but, even then, MAC protocol has to seek the best trade-off between Quality of Service (QoS) and energy efficiency at run time based on the actual status of motes.

In this section, the MAC protocols proposed for EH-WSNs are described briefly along with their fundamental design properties.

2.8.1 Probabilistic Polling for Single-hop WSNs

In [78] a design and analysis of a probabilistic polling algorithm have been proposed. The algorithm exploits the unpredictability of the energy harvesting process to achieve high throughput and fairness as well as low inter-arrival times in EH-WSNs.

Two variants of the CSMA protocol, slotted and unslotted, have been modified for use in EH-WSNs. In the slotted form of the CSMA protocol, there are three states in which a node could be in, the charging, carrier sensing and transmit states. A sensor would only transmit its data packet when the ongoing transmission in the current slot has ended. If there is no transmission in the current slot by any sensor, the sink would transmit a synchronization packet in that slot. A cycle starts when the sensor goes into the charging state and ends when it leaves the transmit state. When the stored energy of the sensor reaches a predetermined amount of energy denoted, it wakes up and goes into the carrier sensing state to wait for the start of the next time slot. At the beginning of the next time slot, it will go into the transmit state and start sending its sensed data to the sink.

For the unslotted CSMA protocol, there are five states in which a sensor could be in, the charging, carrier sensing, receive, idle and transmit states. Initially, the sensor is uncharged so it would be in the charging state. When the energy stored reaches a predetermined amount, it goes into the carrier sensing state to determine whether the channel is free. If the channel is free, it transmits the data packet. Then, it moves into the receive state to wait for an acknowledgement packet. After receiving the ACK packet, it returns to the charging state.

In polling scheme, The sink will transmit a polling packet containing the ID of the sensor to be polled, and the polled sensor will respond with a packet transmission. The polling ID is randomly chosen from the set of all nodes. The authors modified the polling scheme to get a probabilistic one.

In probabilistic polling, nodes first harvest sufficient energy in only a charging state and then stay in a receive state to receive a polling packet. If the level of energy in a node is not adequate to transmit a packet, then the node goes back to a charging state. The distinctive feature of probabilistic polling is that a sink transmits a contention probability, p_c, in a polling packet instead of broadcasting the ID of a sensor. This is to set a probability in each node to decide whether to transmit. When a polling packet is received, the contention probability is compared with a number that is uniformly generated in the range from zero to one. If p_c is greater than the generated number, then the sensor node sends its packet. Upon the reception of the polling packet, it is ideally expected that only one node will be able to transmit a data packet. The p_c is dynamically updated based upon the nodes' responses. If the sink hears nothing after sending the polling packet, then it increases the value of p_c. If a packet transmission is either successful or fails due to a weak signal, then p_c remains at its current value. The value of p_c is reduced when a collision occurs at the sink. Additionally, the value of p_c decreases if new nodes are added to the network, and increases when nodes fail or are removed from the network.

2.8.2 EH-MAC Probabilistic Polling for Multi-hop WSNs

An enhanced version [79] of the probabilistic polling technique discussed above was also proposed for multi-hop communication scenarios common in EH-WSNs. Another solution formulated for the same problem has been presented in this protocol emphasizing on the idea of the number of neighbours currently active for contention probability adjustment. All the nodes taking part in the contention wait for a random time between 0 and t_max and try sending the polling packets only if they sense an idle channel. The polling probability P_c is included in the packets that plays its role in deciding which nodes are eligible for transmission in that specific cycle. Contention probability in this protocol can also be seen as inversely proportional to the number of active neighbours. Moreover, the receiver decreases contention probability where a collision occurs assuming that there are more estimated number of active neighbours than the system is expecting. Similarly, the value of contention probability tends to increase where nodes encounter an empty slot and no one takes part in contention for transmission.

2.8.3 Multi-Tier Probabilistic Polling (MTPP)

MTPP [80] is another protocol with the extension towards achieving multi-hop data delivery that employs a tiered hierarchy model with a cluster of sensor nodes formed based on the distance from the sink. In the single-hop probabilistic polling approach, the sink periodically broadcasts a polling message containing a probability value to all sensor nodes within its transmission range. Since not all sensor nodes are deployed within a communication range of the sink, the sensor nodes outside the communication range have to relay the packet transmission

via other sensor nodes. These intermediate nodes could either respond to the sink or poll further higher numbered tier sensor nodes outside the communication range of the sink. The multi-tier scheme is such that each sensor node belongs to the tier that corresponds to its distance from the sink represented by the number of intermediate hops.

Prior to joining the network, sensor nodes are initialized with a tier number denoting the highest tier number. If it receives any broadcast polling message which has a lower tier number than its own, it assigns its tier to the received tier number plus one. Sensor nodes check if the received broadcast polling message contains a tier number lower than its parent hierarchy (i.e. tier number that is one lower than its current tier) and it immediately updates its tier number if it notices a much lower tier number in the polling message. Conversely, if a node stops receiving broadcast messages from lower numbered tiers but it is able to receive broadcast messages from sensor nodes in the same or higher numbered tiers, then it updates its tier to the received tier number plus one.

A node in a tier should initiate its own polling cycle based on the probability in the polling message it receives that grants it the permission to transmit or not. This means that while the sink polls tier-one sensor nodes, the polled tier-one sensor node either polls tier-two sensor nodes with the same probability or simply transmits its data back to the sink. Each sensor node then stores the data it receives from the higher numbered tier sensor nodes that it has polled, until it gets polled by lower numbered tier sensor nodes. Once a node gets polled, it transmits a data message (Figure 2.6) containing its own data and other data it received from higher tier nodes that it previously polled. The data will be relayed from tier to tier until they eventually reach the sink.

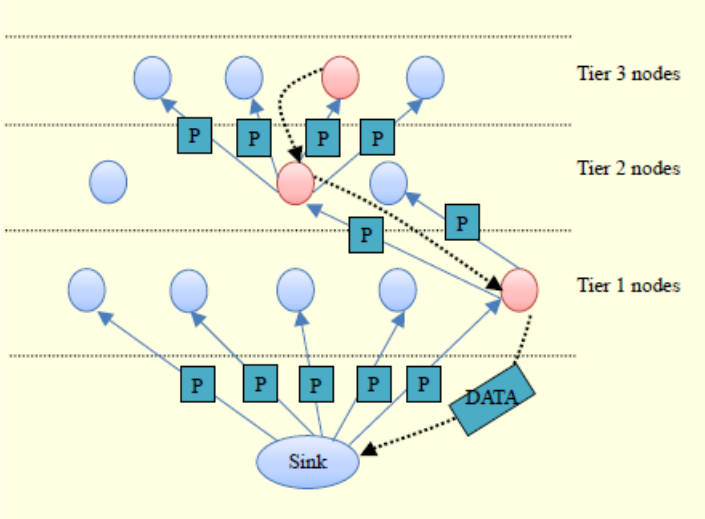


Figure 2.6: MTTP concept with three-tier scale

2.9 Conclusion

The design of MAC protocols for WSNs has been approached assuming battery-powered devices and adopting network lifetime as the primary performance criterion. Therefore, the development of an energy-efficient MAC protocol is one of the major issues in WSNs. Thus, this chapter investigates the state of the art of MAC layer protocols for wireless sensor networks, which includes the design challenges. In particular, a classification of MAC protocols with

energy-efficient characteristics is emphasized. In addition, different MAC protocols for cross-layer and energy harvesting are described.

The next chapter presents a probabilistic evaluation of the standard IEEE 802.11 using stochastic timed automata and statistical model checking.

Chapter 3

Stochastic Generic Model for 802.11 Basic Access MAC Protocol

3.1 Introduction

MAC protocols have gained a lot of importance because of their influence on the lifetime of sensor nodes of WSNs. Major source of energy waste in WSNs is basically collisions. When a transmitted packet is corrupted due to interference, it has to be discarded and the follow on re-transmissions increases energy consumption. Although collisions cannot always be prevented, randomised exponential bake-off rules are used in the retransmission scheme of carrier sense multiple access with collision avoidance (CSMA/CA) to minimise the likelihood of repeated collisions. The complexity of this method and its criticality motivate the formal specification and verification of its basic algorithms. Most existing works do not deal with all possible aspects such as topology, number of nodes, node behaviour, and number of possible retransmissions. In this contribution [81], we propose a stochastic generic model for the 802.11 MAC protocol for an arbitrary network topology which is independent of the number of sensors. In addition to the qualitative evaluation that proves the correctness of the model, we will make a quantitative evaluation using the statistical model checking to measure the probabilistic performance of the protocol.

This chapter describes the primary MAC scheme of the standard IEEE 802.11 and presents the stochastic timed automata which represent the specification of the protocol. In addition, an evaluation of probabilistic performance properties will be exhibit. The chapter ends with a comparison with other works.

3.2 Informal Description

The current section aims to present informally the protocol IEEE 802.11 which will be our case study. The primary MAC scheme of the standard IEEE 802.11 is called Distributed Coordination Function (DCF) [82]. It describes a decentralized mechanism which allows network stations to coordinate for the use of a medium in an attempt to avoid collisions. Three time periods are considered: the DCF Inter Frame Space (DIFS), the Short Inter Frame Space (SIFS) and the EIFS, where SIFS < DIFS < EIFS. A station can start a transmission of data packets only after sensing the medium free for a DIFS. On reception of a data packet, the destination station, after sensing the channel free for SIFS, sends an acknowledgement packet (ACK) back to the sender. A collision is recognised by the sending station if either: on termination of the transmission, the channel is sensed occupied by another station, or if an ACK packet is not received within a given time. In order to avoid collisions, the MAC protocol obliges the stations to enter in backoff stage before sending if either ([1]): *(i)* the channel is not sensed idle for a DIFS, *(ii)* the channel is sensed busy after the station finishes a data transmission, *(iii)* a positive acknowledgement of successful transmission is not received from the destination station before a time-out, *(iiii)* the station receives an acknowledgement and wishes to send another packet.

As soon as a back-off condition becomes true, the deferring station selects a $BackoffTime$ composed of a random number $BackoffValue$ of slot times, where each slot has size $SlotTime$. This value indicates the number of time periods which must be passed before the station can start transmitting. If the channel is detected idle for a some $SlotTime$, the $BackoffValue$ is decremented by 1. This decrementing procedure is temporarily suspended if a transmission is detected (the reduction of $BackoffValue$ is frozen) and is resumed only after the channel is sensed free for DIFS time units. When the $BackoffValue$ reaches 0, the station can start its transmission. The value of $BackoffValue$ is a pseudo-random integer drawn from a uniform distribution over the interval [0, CW], where CW is the Contention Window which

has initial value $CWmin$ and takes values of ascending powers of 2 minus 1. Thus $CW = (CWmin + 1) \times 2^{bc} - 1$ where bc ($BackoffCounter$) increases with the number of consecutive unsuccessful transmissions. Note that the likelihood of a longer back-off delay for repeatedly detected collisions (where bc is large) is increased. The value of CW has an upper bound of $CWmax$. Once this value has been reached, CW will remain at this value until it is reset.

The flowchart of Basic Access (BA) scheme of CSMA/CA algorithm is shown in Figure 3.1.

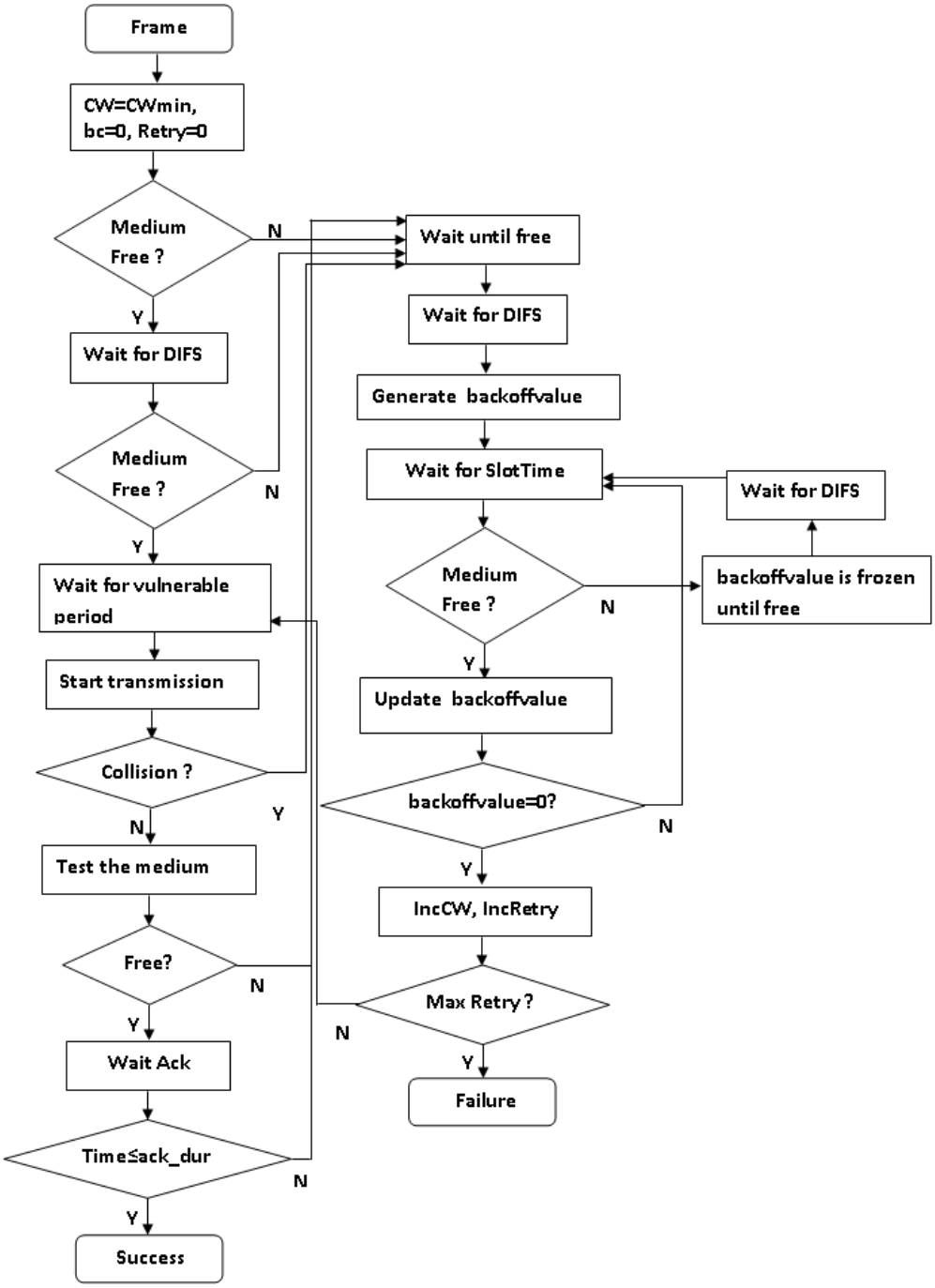


Figure 3.1: Basic Access scheme of CSMA/CA algorithm.

3.3 Formal Tools

This section presents the formal tool used for modelling and verification of the model.

3.3.1 Statistical Model Checking

Statistical Model Checking (SMC) [83] is an innovative approach proposed as an alternative to avoid the exhaustive exploration of the state-space of the model. SMC is a simulation-based solution, which is less time and memory intensive than classical model checking. The idea behind SMC is to generate enough sample execution paths for the system and then to use the statistical hypothesis testing to decide whether the system satisfies the given property or not. Using SMC, one can verify qualitative properties (as in classical model checking) and make quantitative evaluation too. UPPAAL-SMC [84] is an extension of Uppaal, proposed to represent systems via networks of priced timed automata in which probabilistic choices replace non-determinism. Locations are labelled with sojourn time distributions that are *uniform* if invariants are provided, otherwise they are *exponential* with user-defined rates. Likewise, transition edges are associated with weights for the probabilistic choice among multiple enabled transitions. We can now define the stochastic timed automaton used in UPPAAL-SMC as a tuple, $A = (L, L^0, \Sigma, X, I, E, \mu, \omega)$ consisting of a timed automaton $(L, L^0, \Sigma, X, I, E)$ equipped with delay probability density function μ and a probability function ω that assigns a probability to out-put actions [85].

3.3.2 Query Language

UPPAAL-SMC supports five different analysis methods. Below we use N to denote natural number, M to denote number of simulations, P to denote a probability, and exp to denote an expression:

- Statistical evaluation [86]: SMC estimates the probability of the state property being satisfied, using the following query:

$$Pr[<= N](" <> |[]"exp) \tag{3.1}$$

- Hypothesis testing [87]: SMC checks if the property is satisfied within a certain probability, using the following query:

$$Pr[<= N](" <> |[]"exp)" <= | >= "P \tag{3.2}$$

- Statistical comparison: SMC compares the satisfaction possibilities over two properties, using the following query:

$$Pr[<= N1](" <> |[]"exp1)" <= | >= "Pr[<= N2](" <> |[]"exp2) \tag{3.3}$$

- Expected value: SMC computes the maximal or the minimal value of a certain variable while checking the system, using the following query:

$$E[<= N; M](" min|max" : exp) \tag{3.4}$$

- Simulations: SMC simulates a system multiple times and computes trajectories of specified expressions over time, using the following query:

$$simulate\ M[<= N]\{exp1, exp2\} \tag{3.5}$$

3.4 Stochastic Models of the Protocol

In this contribution, we developed stochastic generic model for the 802.11 basic access MAC protocol. The model consists of two synchronized stochastic timed automata: Wireless station (WS) model, and Medium Model.

The model of a station represents the sending and receiving behaviour of a station as shown in Figure 3.2. The stochastic timed automaton begins in $Idle\_listening$, if the medium is free and the station has a frame to send, then the station starts sensing the medium. If the medium remains free for DIFS, the station enters the $vulnerable$ state, where it switches its transceiver to transmit mode and begins transmitting the frame with broadcast synchronization channel ($sent$). Otherwise the station enters back-off phase. The transmission may be terminated successfully or with a collision. If so, the station will immediately be in the $test\_channel$ state. If the medium is free, it must wait for the acknowledgement time-out ack_dur (to let the transmission in progress finishes) then it enters the back-off. On a successful transmission, the station waits for ack_dur. If it does not receive an ACK during this time, it activates the back-off algorithm, otherwise the frame has been sent correctly and the station moves to the $wait\_pdu$ state. The station stays in this state for EIFS before trying to transmit another packet.

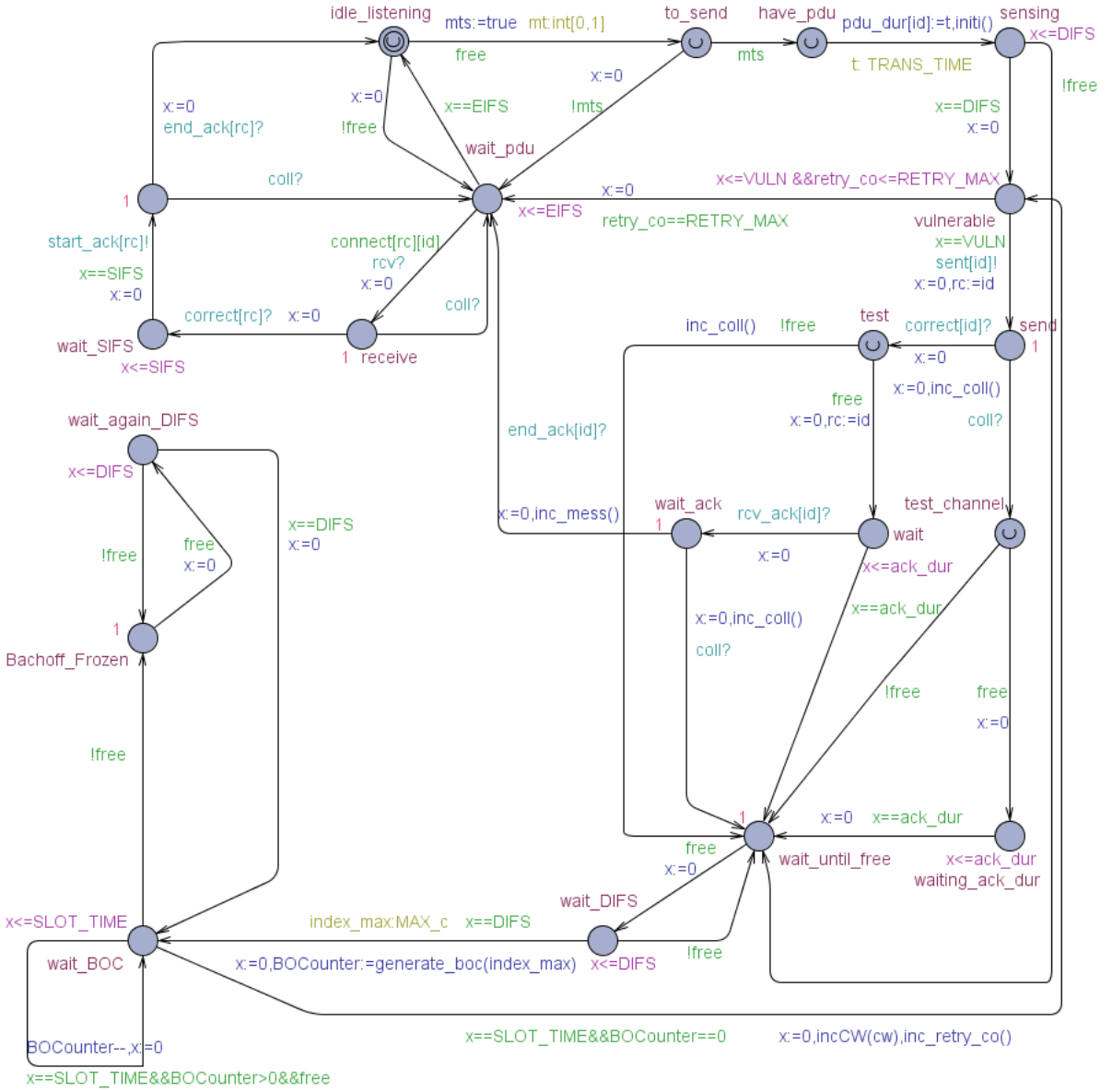


Figure 3.2: Stochastic timed automaton of the station.

In the back-off phase, the station waits for the medium to be free for DIFS and then generates a random back-off value. After $SLOT\_TIME$, if the medium remains free, the station decrements the back-off value by 1, otherwise it is frozen until the medium becomes free. When the back-off value reaches 0, The number of retransmissions($retry\_co$) will be incremented by 1 as well as the value of bc and the cw will be increased. Then the station can restart the transmission of the frame if $retry\_co$ does not reach the maximum number of attempts($RETRY\_MAX$), if not it moves to the $wait\_pdu$ state.

In the $wait\_pdu$ state, the station changes the behaviour and becomes a receiving station. It remains in this state for an EIFS in order to receive messages or to release messages which can be blocked waiting for this station. If the station received a message from one of these neighbours (the guard $connect[rc][id]$ where id is a unique number that represents the station and rc is a unique number that represents the sender), it waits for a SIFS before sending an acknowledgement.

The stochastic timed automaton of Figure 3.3 which represents the medium starts with the initial state $idle$. This state means that the medium is free. On receiving a packet from a station n, the medium immediately broadcasts it to the other stations. The transmission time is non-deterministic, it is within $TRANS\_TIME\_MIN$ and $TRANS\_TIME\_MAX$ which are the minimum and maximum allowable packet transmission times respectively. The transmission can be terminated correctly or with a collision. If the medium has received an acknowledgement, it sends it immediately to its recipient. The same, this transmission can be terminated with a collision.

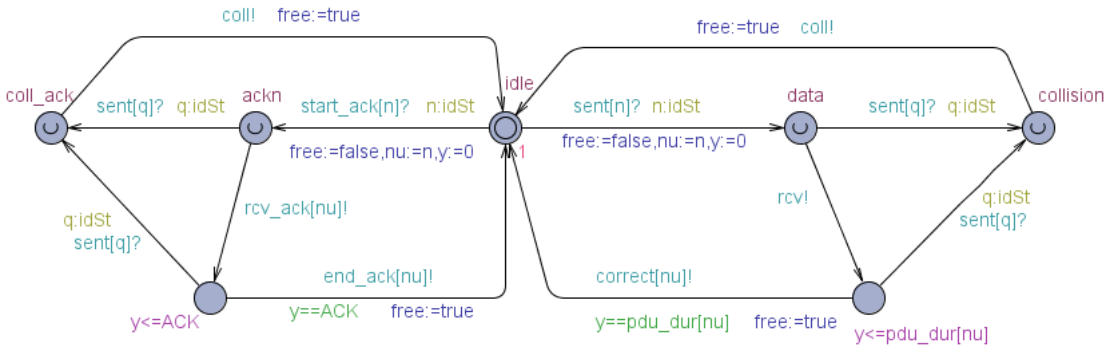


Figure 3.3: Stochastic timed automaton of the medium.

3.5 Protocol Analysis

In Uppaal, the delay in a state is either:

- 0 if the state is urgent or committed;
- A chosen value with a uniform distribution in the presence of the invariant;
- A chosen value with an exponential distribution (we use a rate of 1) in the absence of the invariant.

The WSN to be verified contains NB_WS sensors. It is a priced timed automata network consisting of NB_WS priced timed automata that synchronize with the medium. The topology of the WSN is with full connectivity, all sensors are neighbours with each other.

To evaluate the probabilistic performance properties, we use the statistical model checking incorporated within Uppaal. The correctness verification of the design of the protocol can be performed with qualitative properties, and the performance evaluation can be done with quantitative properties.

In this analysis we use the parameters of the Frequency Hopping Spread Spectrum (FHSS) physical layer as shown in Table 3.1.

We then applied time-scale abstraction to the model by scaling the timing values by the $SLOT\_TIME$ to minimize run time.

We use also an adjacency matrix $connect[NB\_WS, NB\_WS]$ (NB_WS represents the number of the stations) that represents the network topology which is a Boolean matrix, 1 if the stations are neighbours and 0 if not. We suppose also that maximum number of retransmission ($RETRY\_MAX$) is 5.

Table 3.1: Parameter values

Variable	***Value***
SLOT_TIME	50μs
SIFS	28μs
DIFS	128μs
EIFS	361μs
VULN	48μs
TRANS_TIME_MIN	224μs
TRANS_TIME_MAX	15717μs
ack_dur	300μs
ACK	205μs
CW_MIN	15
CW_MAX	1023

3.5.1 Verification of Qualitative Properties

We can check functional properties of the protocol, such as whether there are deadlocks, whether all nodes can send messages, and whether nodes succeed in receiving acknowledgements.

- *Message sending.* Check whether it is possible for all nodes in the network to send messages at all and therefore they will pass to waiting acknowledgement. This property can be specified by a CTL formula:

$$E <> forall(i : idSt)sensor(i).wait \tag{3.6}$$

- *Message receiving.* Check whether any message that has been sent should eventually be received. This property is checked by:

$$A <> forall(i : idSt)sensor(i).wait\_Ack \tag{3.7}$$

- *Deadlock Free.* Check whether the model is deadlock-free using the formula:

$$A[]notdeadlock \tag{3.8}$$

The first property is always verified whatever the number of stations, the second is not verified because a station can enter into collision paths and never complete its transmission and the third is not decidable when the number of stations increases.

3.5.2 Verification of Quantitative Properties

We can estimate the probability that a certain property will be satisfied by the system prior to the violation of some constraint. In other words, we are not interested here in whether a station reaches a state or not, but to know with what rate this station will reach this state.

- *Probability of sending.* Computes the probability that a station reaches a state where it has finished its transmission and waits for an ACK before 1000 units of time during the run of the protocol. This probability can be estimated with the PCTL formula:

$$Pr[<= 1000](<> sensor(0).wait) \tag{3.9}$$

- *Probability of receiving.* Computes the probability that a station reaches a state where it starts receiving an ACK before 5000 units of time during the run of the protocol.

$$Pr[<= 5000](<> sensor(0).wait\_ack) \tag{3.10}$$

- *Probability of termination.* Which allows to know the probability that all stations finish the transmission correctly.

$$Pr[<= 5000](<> forall(i : idSt)sensor(i).wait\_pdu) \tag{3.11}$$

The verification results of these probabilities are shown in Table 3.2.

Table 3.2: Probabilities intervals of sending, receiving, and termination

Number of nodes	***Probability 1***	***Probability 2***	***Probability 3***
2	[0.90,1]	[0.90,1]	[0.81,0.91]
3	[0.90,0.99]	[0.62,0.72]	[0.60,0.70]
4	[0.82,0.91]	[0.46,0.56]	[0.52,0.62]
5	[0.77,0.87]	[0.40,0.50]	[0.37,0.47]
10	[0.41,0.51]	[0.14,0.24]	[0.02,0.12]
20	[0.21,0.31]	[0.03,0.13]	[0,0.09]

For this verification we assumed the worst case where all the stations have messages to send ($mts = true$), initially all stations collide. We use also the same deadline for all cases. From the table, we see that the probabilities are degrading with the increase in the number of nodes. This result is reasonable because the number of collisions increases with the number of nodes. To always have (whatever the number of nodes) a high probability it is necessary to find the appropriate period for each case. From several experiments, we found the time required in each case as shown in Table 3.3.

Table 3.3: Deadline on number of nods

Number of nodes	***Deadline***
2	1000
3	1500
4	1500
5	2500
10	7000
20	7000

It is also possible to evaluate the expected properties as follows:

- *The average of the maximum number of messages to sent.* After 36 runs, this average is estimated to be in the confidence interval 23.61 ± 1.40 within the first 5000 time-units: $E[<= 5000; 36](max : nbt\_mtos)$. The frequency histogram in Figure 3.4 shows this result.

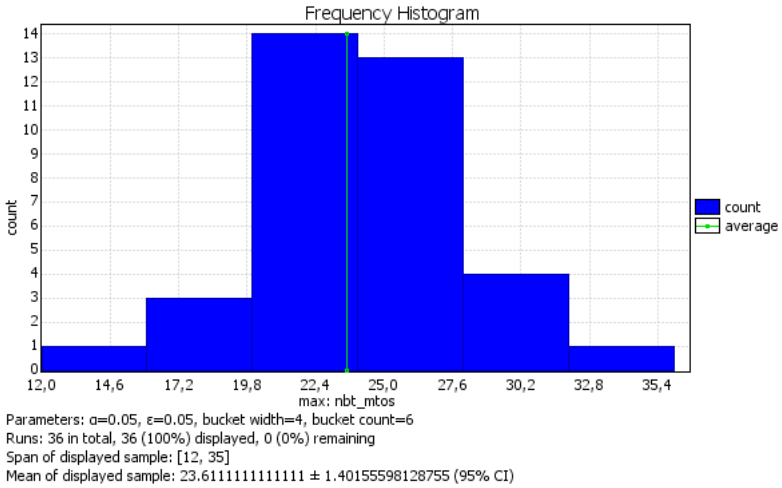


Figure 3.4: Frequency histogram of maximum number of messages to sent.

- *The average of the maximum number of messages effectively sent.* After 36 runs, this average is estimated to be in the confidence interval 20.5 ± 1.39 within the first 5000 time-units: $E[<= 5000; 36](max : nbt\_mes)$. Figure 3.5 shows the result.

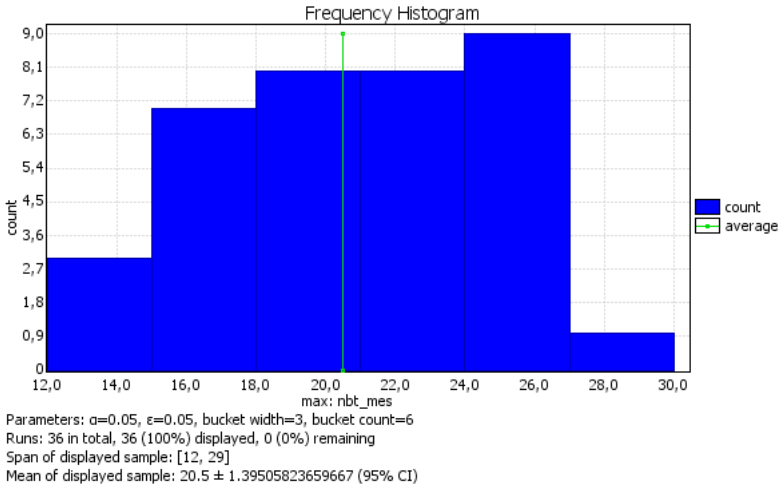


Figure 3.5: Frequency histogram of maximum number of messages effectively sent.

- *The average of the maximum number of collisions.* After 36 runs, this average is estimated to be in the confidence interval 2.5 ± 0.29 within the first 5000 time-units: $E[<= 5000; 36](max : nbt\_coll)$. the result of this verification is shown with the frequency histogram in Figure 3.6.

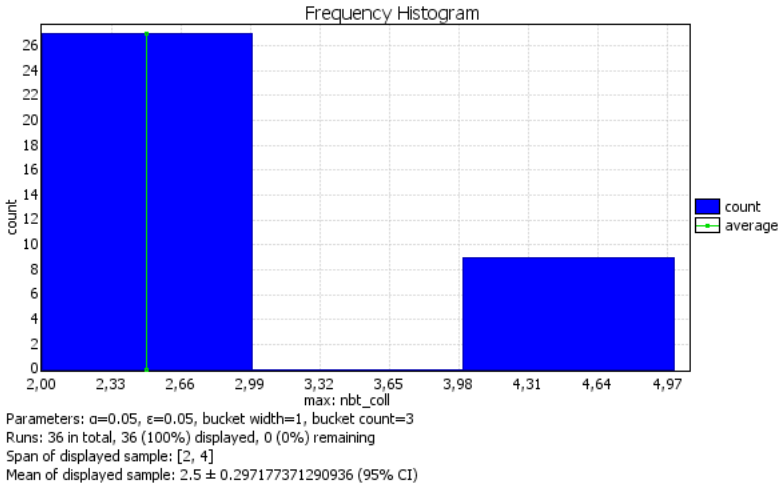


Figure 3.6: Frequency histogram of maximum number of collisions.

The results show that the number of collisions is small compared to the number of messages actually sent, which proves the validity of the model.

3.6 Comparison

To better show the strengths of our contribution, we propose to make a comparison with previous works (see Table 3.4).

Table 3.4: Comparison with previous works

Works / Attributes	[1]	[2, 3]	[88]	[5]	Our Model
Number of stations	2	2	3	-	unlimited
Behavior of the station	send	send	send	send/receive	send/receive
Collision of acknowledgment	No	No	No	Yes	Yes
Extended InterFrame Space	No	No	No	Yes	Yes
Number of retransmissions	infinite	infinite	infinite	infinite	5
Topology of the network	No	No	No	No	Yes

In [1], the authors presented a formal study of a two-way handshake sub-protocol of the IEEE 802.11 standard. They studied a case of two sending stations and two destination stations. They made three hypothesis: (i) there is not an EIFS (Extended Inter-frame Space) parameter, (ii) there is no timing synchronization function, (iii) and finally, the retry limits is supposed to be infinite. The work specified the protocol using probabilistic timed automata and expresses probabilistic reachability properties with PCTL logic. The probabilistic choice, in this work, is presented in the randomized backoff procedure. In [2, 3], the network consists of two stations. Models for slotted and non-slotted CSMA-CA in IEEE 802.15.4 protocol are developed using

probabilistic timed automata. A range of performance measures to different scenarios are applied. The authors evaluated probabilistic reachability properties and some expected reachability properties. The authors of [88] developed generalized probabilistic timed automata models of the 802.11 basic access MAC protocol, that are independent of the number of stations. They applied a number of optimizations and a set of reductions that reduce the generalized multi-station model. They succeed to model-check a topology of three stations. The authors of [5] proposed a new variant of the CSMA/CA with DCF mode. In this variant, each station has to disconnect whenever its signal-to-noise ratio is lower than a specific threshold. Such disconnections are intended to reduce the number of collisions and to improve the transmission rate. In this work, the authors proved the absence of a deadlock and the successful termination of a transmission.

3.7 Conclusion

CSMA/CA technique is one of the most used in MAC layer for WSNs. This technique introduces random back-off based procedure to avoid the increase of collisions when the number of nodes increases. Many works have proposed extensions, adaptation and studies of CSMA/CA in WSNs. The use of formal methods in this field is an attractive and promising issue. Formal methods allow high level specification, verification and evaluation of protocols and techniques.

In this chapter, we have proposed a stochastic generic model for the 802.11 basic access MAC protocol which is independent of the number of stations. The model represents the two behaviours, sending and receiving, of the station. Indeed, we have developed a model where we have incorporated all possible parameters that can influence protocol performance such as the network topology, the number of possible retransmissions, the behaviour of the node as well as the number of nodes in the network. In addition, the use of priced timed automata for modelling ensures the proper simulation of the network. The delays in states are not the same. They depend on the invariant and the nature of the state(urgent or committed or not). To verify and evaluate the performance of the protocol, we have used statistical model checking in the Uppaal tool. Properties that guaranty the correctness of the protocol(e.g., reachability, liveness, deadlock Free) were verified. Also, some important metrics were evaluated(e.g., probability of success, throughput, number of collisions). The results prove the correctness, the efficiency and the scalability of the model.

The following chapter will be directed towards another source of energy for wireless sensor networks, that of ambient energy. Therefore, the next chapter will be devoted to a formal study of one of the protocols designed for WSNs powered by ambient energy.

Chapter 4

Performance Evaluation of ODMAC Protocol for WSNs Powered by Ambient Energy

4.1 Introduction

Designing a good MAC protocol always remains a challenge. A good one in the sense that it grants access to the medium (avoids collisions) on the one hand and minimizes energy consumption (minimizes idle listening of sensors) on the other. Nevertheless, and with the appearance of Energy Harvesting-Wireless Sensor Networks (EH-WSNs), energy is no longer a problem (under normal conditions) but the challenge now is that each sensor remains in its energetically sustainable state (the consumed energy is always lesser than the harvested energy) as much as possible.

In this chapter, we focus on a protocol designed for EH-WSNs networks, "On Demand MAC (ODMAC)". First, we will present the algorithms showing the steps to follow when packets are sent or received. Then, we will elaborate formal models, as stochastic timed automata, that represent the network entities involved in ODMAC: the sender, the receiver and the channel using Uppaal tool. In addition, we will develop three other models, one to represent the cycle adjustment behaviour of a node, the second to monitor the battery status of a node and the third to represent the energy harvesting rate of a node. Finally, the protocol verification is carried out using the query language PCTL and TCTL provided in UPPAAL-SMC. This verification concerns various parameters such as delay, sensing rate, energy harvesting rate, energy consumption and the ratio HCR (harvested energy over consumed energy).

4.2 Related Work

The requirement of low power consumption is a persistent constraint for sensor nodes that rely on portable batteries with limited capacity as a power source. Energy-efficient communication protocols can extend network lifetime to the detriment of the quality of the services provided. Therefore, finding an alternative power source to operate WSNs becomes imperative. Energy harvesting is a promising technology that allows nodes to renew their energy from an ambient energy source (solar power, mechanical vibrations, wind, etc.) [89]. Indeed, sensor nodes are able to convert harvested energy from the surrounding environment sources into electricity to power themselves. Evolution of energy recovery technologies has led to the development of EH-WSNs. Each sensor node of the EH-WSNs is further equipped with energy harvesters and a storage capacitor to accumulate the recovered energy [90]. However, the energy harvesting rates are significantly lower than the energy consumption for node operation, so that, the capacitor(or supercapacitor) stores energy until it attains a certain level sufficient to operate the node. Fortunately, with storage devices having an almost unlimited number of recharge cycles, EH-WSNs can work for a long time without having to manually refill their power [78].

Medium access control protocols still play an interesting role in the design of WSNs. In EH-WSNs, the major challenge is that the charging time of sensor nodes to a sufficient level varies due to various factors including, but not limited to, environmental factors and the type of energy harvesting devices used. Many energy-aware MAC protocols have been proposed for WSNs, aiming to minimise power consumption and to extend network lifetime through minimizing packet collisions, idle listening, overhearing and especially turning off the radio as much as possible. However, they are not upgraded for the energy characteristics of EH-WSNs where the objective is to maximize network performance through efficient use of the harvested energy [91].

A node is in Energy Neutral Operation (ENO), if the required performance level is guaranteed while ensuring that the node never fails due to power exhaustion [90]. Thus, a node should always control its energy consumption so that it does not exceed the harvested amount (i.e. it should consume less than what is harvested). A node enters the ENO-MAX state when it is able to reach ENO state with maximum performance. Therefore, MAC protocols designed for EH-WSN networks must support the node achieving the ENO-MAX state.

In this context, few works have studied MAC protocols on EH-WSNs. Among the surveys that have been done, we can cite the work of [92]. In this last survey, the authors presented a review of MAC protocols designed for EH-WSNs, then a comparison was done between these protocols based on different performance metrics such as throughput, fairness, scalability and latency. More detailed study on energy harvesting challenges was presented in [77] where the authors gave the technical aspects of energy harvesting technologies, then they provided a classification taxonomy of special MAC protocols for EH-WSNs protocols. Furthermore, the pros and cons of each protocol are thoroughly analysed.

In [78], the authors evaluated the performance of CSMA and polling based MAC protocols when used in EH-WSNs in a one-hop scenario. Furthermore, they designed a probabilistic polling protocol with a variable that represents the probability of contention in order to manage collisions and harvesting rates. These analyses are validated through simulation where the evaluation criteria considered are throughput, fairness and inter-arrival time. Eventually, a comparison and evaluation of various MAC protocols is established. The work presented in [93] assessed the ability of MAC protocols specifically TDMA, Framed-ALOHA (FA) and Dynamic-FA (DFA) to provide measures of any sensor to its destination for single-hop EH-WSNs. The analytical derivations are investigated using Markov models and validated through numerical simulations.

The authors of [94] proposed a new MAC scheme, called On Demand MAC (ODMAC), designed specially for EH-WSNs. ODMAC is an asynchronous protocol which supports individual duty cycles. Thus, each node adjusts its duty cycle to converge as closely as possible with the ENO-MAX state. Moreover, two performance metrics, namely the packet delay and the sensing rate, were evaluated through simulations using the OPNET simulator. In their next work [95], the main objective was to make a comparison between two asynchronous MAC protocols, ODMAC and a basic version of XMAC [96] and to discuss which one of them is more suitable for EH-WSNs. Through their analysis, they evaluated the protocols in terms of energy consumption overhead and channel utilization overhead. In order to validate the theoretical results, the same authors proposed an implementation of ODMAC protocol in [97], where the throughput of the application was tested at different levels of input power. Further, [98] implemented and evaluated ODMAC using an off-the-shelf micro-controller and an off-the-shelf photovoltaic energy harvester.

4.3 Basic MAC Schemes Evaluation

A variety of MAC protocols have been proposed for WSNs [99, 54] but they are unable to meet the requirements imposed by EH-WSNs. In MAC protocols designed primarily for WSNs known as duty cycle protocols, each node has two cycles: sleep and wake. In the sleep cycle, sensor nodes turn off their radio to conserve energy. The nodes wake up on the same schedule (synchronous) or each one has its own schedule (asynchronous) [100]. In synchronous schemes, nodes have the same wake up time and they exchange data packets and synchronisation requests during the active cycle. One of the most cited protocols in this category is S-MAC [46]. Among these protocols, we find also: T-MAC [59], R-MAC [101] and DSMAC [102]. On the other

hand, each node in asynchronous schemes has its independent schedule. B-MAC [57] uses the sender-initiated paradigm, in which the receiver stays in idle listening state waiting for incoming preambles. In contrast, RI-MAC [103] uses the receiver-initiated paradigm, in which the sender remains waiting to receive beacons from the receiver.

Synchronous schemes use a common sleeping schedule where nodes waking up simultaneously. A node in charging state cannot wake up at the time, hence this mechanism is not suitable for EH-WSNs. Instead, asynchronous schemes are more suitable for EH-WSNs because they use independent duty cycles.

4.4 On Demand Medium Access Control (ODMAC)

ODMAC [94] has especially been designed for the realization of EH-WSNs; it is an asynchronous MAC protocol which follows the receiver-initiated paradigm. The receiver wakes up periodically and sends beacons indicating that it is available to receive data. Whenever, the sender becomes active, it continues listening to receive the appropriate beacon. As soon as the sender receives this beacon, the transmission would instantly be started.

In ODMAC, a node can only perform one operation at a time, either sending or receiving. For that, a node has two different periods: the beacon period and the sensing period. The beacon period determines the duty cycle of packet retransmission while the sensing period determines the duty cycle of the data measuring. In order to achieve ENO state (or ENO-MAX in the best case), ODMAC must properly control the amount of the available energy at a given time. This power control (energy spending or conservation) affects system performance (end-to-end delay and throughput). Hence the beacon period balances between end-to-end delay and energy consumption. When the beacon period is high, the less beacons are transmitted and the higher the energy conservation is. Therefore, an increase in the end-to-end delay of the packet. On the other hand, the sensing period balances between throughput and energy consumption.

When a receiver wakes up, it listens to the channel for T_{IFS} time units. If the channel is free, the receiver transmits the beacon and continues to wait for incoming packets for T_{TX} time units. If no packet is received during this period, it returns to its sleep state. It is necessary that T_{IFS} must be greater than T_{TX} in order to avoid that the node transmits a beacon while another is waiting the reception of data. On the other side, after receiving the beacon, the sender is delayed a random number of time units, T_{slot}, within the interval specified by the contention window (CW). If the channel is free, the sender transmits its data packet and returns to a sleeping state.

We propose the following algorithms which summarize the communication between the sender and the receiver.

Algorithm 1 Receiver side

```
s:
time ← 0
while time ≤ beacon_period do
    time ← time + T_slot
end while
listen_channel_T_IFS
if channel_free() then
b:
    transmit_beacon()
    time ← 0
    while time ≤ T_TX ∧ ¬receive_data() do
        time ← time + T_slot
    end while
    if ¬receive_data() then
        go to s                                   ▷ go to sleep
    else
        go to b                                   ▷ retransmit a new beacon
    end if
else
    go to s                                       ▷ go to sleep
end if
```

Algorithm 2 Sender side

```
s:
time ← 0
while time ≤ sensing_period do
    time ← time + T_slot
end while
l:
time ← 0
while time ≤ waiting_time ∧ ¬receive_beacon() do
    time ← time + T_slot
end while
if ¬receive_beacon() then
    go to s                                       ▷ go to sleep
else
    TB ← T_slot * int[0, CW − 1] // delays a random number of time units within the interval
specified by the contention window
    listen_channel_TB
    if channel_free() then
        transmit_data()
        if more_queued_packets() then
            go to l                               ▷ go to listening
        else
            go to s                               ▷ go to sleep
        end if
    else
        go to l                                   ▷ go to listening
    end if
end if
```

The formal algorithmic description of the protocol presented in this section is derived directly from the informal descriptions of the original papers [94] and [97].

As we have already explained, a node must remain in ENO-MAX state as much as possible. Therefore, a node with a high harvesting rate can reduce one of its duty cycles, whereas a node with a low rate may increase one of these duty cycles according to the needs of the application.

Periodically, a node calculates the ratio of the harvested energy to the consumed energy. If this ratio is greater than a certain threshold, then the node has a surplus of energy that it can use to optimize its performance. If the throughput is the most important performance measure ($SProb = 0.75$), then the node decreases the sensing period. Otherwise, if the delay is the most important ($SProb = 0.25$), then the node decreases the beacon period. On the other hand, if the node has a lack of energy, then it increases one of the two periods depending on the value of $SProb$.

We have formalized this behaviour as the following algorithm.

Algorithm 3 Duty cycle adjustment

$HCR \leftarrow Harvested\_energy/Consumed\_energy$
if $HCR > threshold$ **then** ▷ extra harvested energy
  if $SProb = 0.75$ **then** ▷ favour the throughput
    $decrease\_sensing\_period()$
  end if
  if $SProb = 0.25$ **then** ▷ favour the end-to-end delay
    $decrease\_beacon\_period()$
  end if
else ▷ deficient energy
  if $SProb = 0.75$ **then**
    $increase\_beacon\_period()$
  end if
  if $SProb = 0.25$ **then**
    $increase\_sensing\_period()$
  end if
end if

4.5 Performance Metrics

4.5.1 Delay and Throughput

Delay and throughput are important metrics to evaluate MAC protocols. The end-to-end delay refers to the time needed for the destination to get the packet generated by the source. It comes from several sources including transmission delay, propagation delay, synchronization delay, processing delay and queuing delay. We consider here the transmission delay and the synchronization delay. The transmission delay is equal to $\frac{L}{R}$, where L is the size of a packet in bits and R is the rate of transmission in bits per second. The synchronization delay is the time spent in idle listening ($waiting\_time$) before receiving a beacon.

The throughput refers to the amount of data reaches the destination effectively. The quantity of packets generated by a sensor indicates the sensing rate (r) which is equal to $\frac{1}{s}$ where s is the sensing period.

4.5.2 Power Consumption

Since the transceiver is the most power-consuming element of a wireless sensor, we are interested in this work to the energy consumed in communications. Four possible modes to operate the transceiver are : *Sleep*, *Idle*, *Transmit* and *Receive*. In the following, we give formulas for the consumed power in transmission and reception.

1 consumed power for transmitting packets (P^{tx}) is given by equation (4.1). P^t is the power consumed while transmitting, r is the sensing rate of the node, L is the packet size, and R is the transmission rate.

$$P^{tx} = P^t r \frac{L}{R} \tag{4.1}$$

2 consumed power for receiving packets (P^{rx}) is given by equation(4.2), such that P^r is the power consumed while receiving and r^f is the traffic rate of the forwarded packets.

$$P^{rx} = P^r r^f \frac{L}{R} \tag{4.2}$$

3 consumed power for transmitting beacons (P^{tb}) is given by equation(4.3), such that t is the beaconing period and L_b is the beacon size.

$$P^{tb} = P^t \frac{1}{t} \frac{L_b}{R} \tag{4.3}$$

4 consumed power for receiving beacons (P^{rb}) is given by equation(4.4) where y is the waiting time for a beacon.

$$P^{rb} = P^r y r \tag{4.4}$$

More details on the above metrics can be found in [104] and [105].

4.6 Modelling ODMAC using Timed Automata

In this section, we present six models which represent the different behaviours of entities participating in communication. These models are developed using stochastic timed automata formalism described in the previous section. Figures 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 show the six models.

The receiver starts with $sleep$ state. In this state, any packet received is discarded. Whenever the beacon period expires, a Clear Channel Assessment (CCA) for T_{IFS} time units is performed. If the channel is busy, the sensor returns to the $sleep$ state. Otherwise, it transmits a beacon and goes to $wait\_data$ state in which it waits to receive packets for a pre-set T_{TX} period. If this period is over without receiving any packets, it returns to the $sleep$ state. After a successful packet reception, the receiver retransmits a new beacon (see Figure 4.1).

Similarly, the sender (shown in Figure 4.2) begins with $sleep$ state. While being in the sleeping state, a new packet was generated. The sensor moves to $wait\_beacon$ state in which it waits for a beacon. After the beacon reception, the sender goes to $backoff$ state. If the channel is free, the sender transmits the packet and returns to $sleep$ state if there is no packet to send yet. Unless, it goes back to the $wait\_beacon$ state.

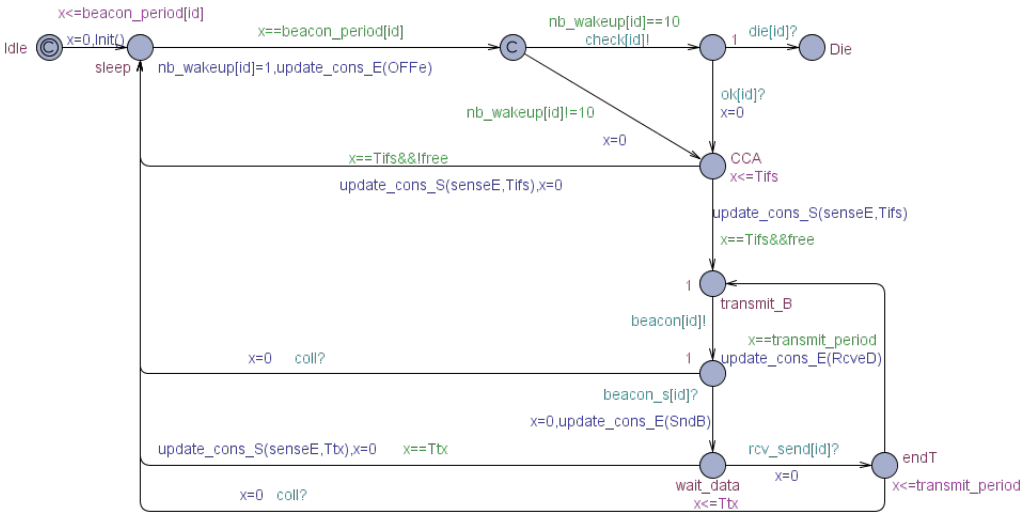


Figure 4.1: Model of the receiver

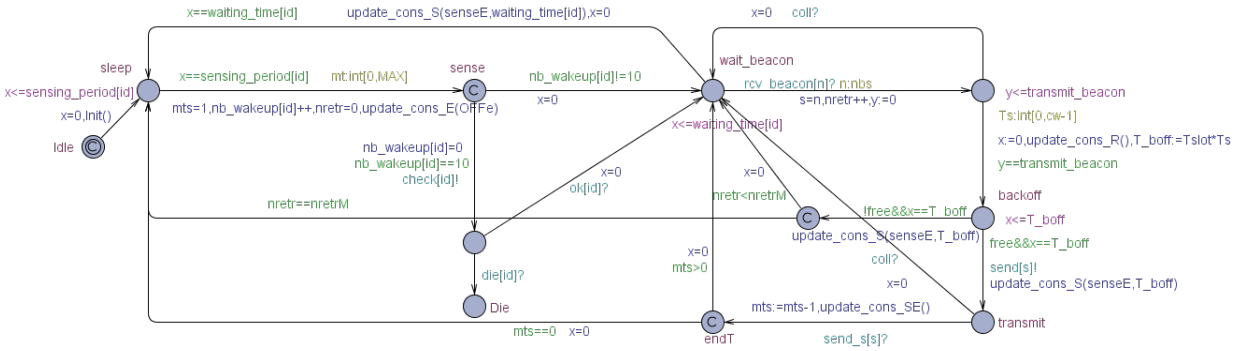


Figure 4.2: Model of the sender

The medium (shown in Figure 4.3) manages the transmission of data packets and beacons between a sender and a receiver. This transmission can be achieved normally or with a collision.

To achieve ENO-MAX sate that minimizes the consumed energy and maximizes the performance, ODMAC periodically adapts the two duty cycles (beacon period and sensing period). To represent this behaviour, we added another timed automata that we have called 'adapter' as shown in Figure 4.4. An adapter is associated with each sensor (sender or receiver). After the expiration of a certain period ($adapt\_time$), a ratio ($ratioHC$) of the harvested energy over the consumed energy is measured. If this ratio is outside a certain determined interval, the cycle is adjusted. The adjustment step is controlled by $SProb$ parameter as explained in section 4.4.

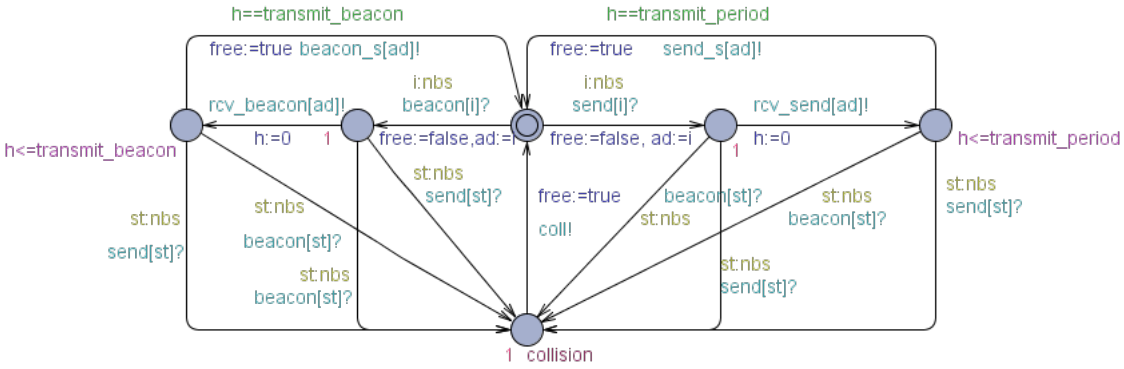


Figure 4.3: Model of the channel

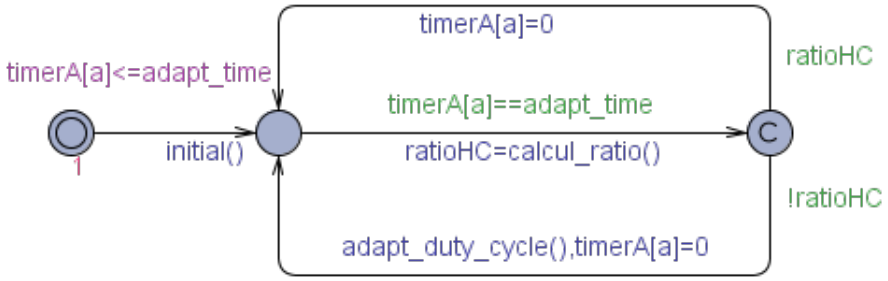


Figure 4.4: Model of the adapter

The model of Figure 4.5 checks the battery status every 10 wakes up. If the energy is below a minimum threshold for transmission, then the sensor goes to charging (this function represents the energy harvesting). After the charging time has elapsed, the energy will be measured once again. If it remains minimal, the sensor will die. This is important in case where the sensor is far or hidden from the energy source.

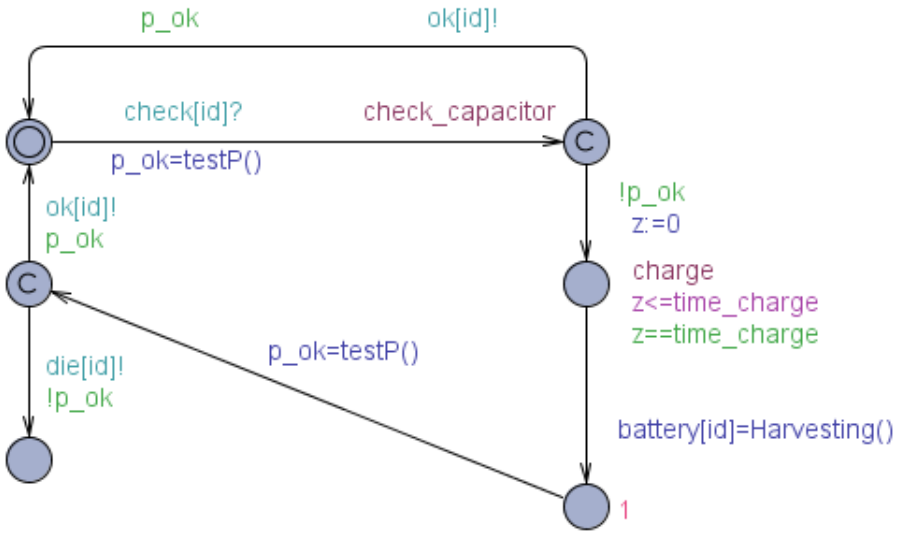


Figure 4.5: Model of the capacitor

Finally, the energy harvesting rate of each node is modelled as random variable that follows a normal distribution. The model of the harvester is given in Figure 4.6.

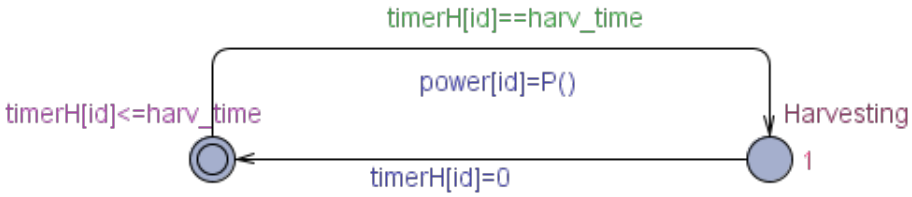


Figure 4.6: Model of the harvester

4.7 Analysis using UPPAAL-SMC

We opt to analyse the ODMAC model using the Uppaal model-checker. Performance evaluation is realised using the query language TCTL and PCTL provided in the Uppaal SMC. The protocol parameters used in the experiments are depicted in Table 4.1. The values of the battery and energy consumption parameters are taken from [105].

Table 4.1: Parameters used in the analysis

Parameter	*Value*
Data rate	$256kbps$
Message length	$100bytes$
Beacon size	$8bytes$
Duration of transmission of data packet	$3.125ms$
Duration of transmission of beacon packet	$0.25ms$
T_{TX}	$66ms$
T_{slot}	$1ms$
CW	63
T_{IFS}	$70ms$
Charging_time	$10s$
Initial energy	$560mA$
Transmission power	$10dBm$
Power consumed while transmitting	$26.7mA$
Power consumed while receiving	$9.2mA$
Power consumed while Idle	$6mA$
Power consumed while sleep	$1\mu A$

In the following, three protocol evaluations will be carried out. The objective of the first evaluation is to study the impact of the beacon period and the sensing period on energy consumption. Where as the second evaluation aims to achieve the ENO-MAX state by adjusting

sensing period and beacon period. The last evaluation shows the effect of changing the harvesting rate on the behaviour of the node.

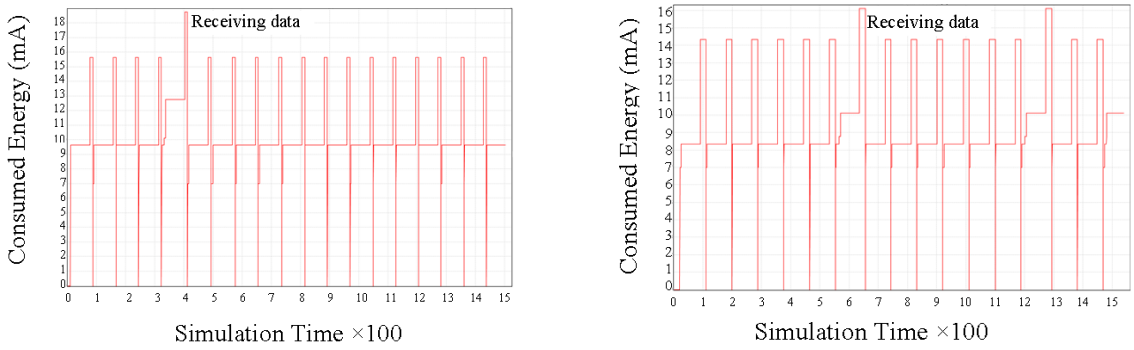
4.7.1 Static Duty Cycle

The first evaluation demonstrates the basic properties of the protocol. The adjustment step is deactivated (i.e. beacon period and sensing period are statically predefined). We start this evaluation with one-hop topology which consists of one transmitter and one receiver. The beacon period and sensing period are fixed at $20ms$ and $60ms$ respectively. We also take $the\ harvesting\ rate$ as $40mA/cycle$ and $waiting\_time = beacon\_period/2$, for a reason of simplicity.

The first constraint in the protocol is to define the right timing parameters to ensure that a transmitter is active when a receiver has sent a beacon. We check if the sender arrives to send his packet in an appropriate time with the formula: $Pr[<= 1500](<> sender0.endT)$. This shows the probability that the sender reaches this state with a runtime lower or equal to 1500 units of time. In the same way, we check if the receiver arrives to send the beacon and receives the packet with the formula: $Pr[<= 1500](<> receiver1.endT)$. For both properties, we have a good result with probability interval $[0.9, 1]$.

4.7.1.1 Consumption and Duty Cycles

Since a node in ODMAC has two duty cycles, a beacon duty cycle and a sensing duty cycle, increasing the beacon period decreases the power consumption connected to the beaconing process. Figures 4.7.a and 4.7.b show the degradation of energy consumption for successive beacon periods $10ms$, $20ms$ while the sensing period is fixed to $60ms$. These figures are obtained by executing the query $simulate1[<= 1500]receiver1.c\_energy$.

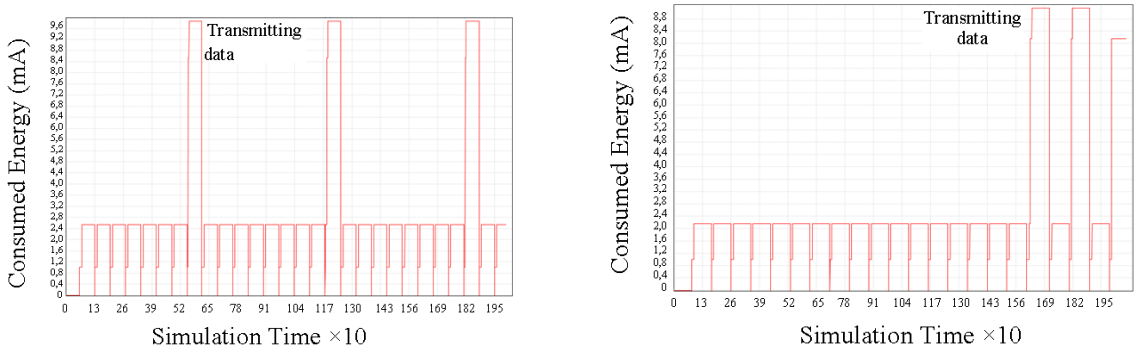


(a) beacon period=10ms (b) beacon period=20ms

Figure 4.7: Consumed Energy for different beacon periods.

The sensing period also affects the energy consumption. A decrease in the sensing period leads to an increase in the sensing rate and an increase in energy consumption. Figure 4.8 depicts the energy consumption of the sensing node for the sensing periods $60ms$ and $80ms$ while the beacon period is fixed to $20ms$. The figure is the result of the query : $simulate1[<= 2000]sender0.c\_energy$.

From these results, we also note that the receiver consumes more energy than the transmitter (which is not really the case) because of synchronization. The receiver loses a beacon each time until the transmitter wakes up.



(a) sensing period=60ms (b) sensing period=80ms

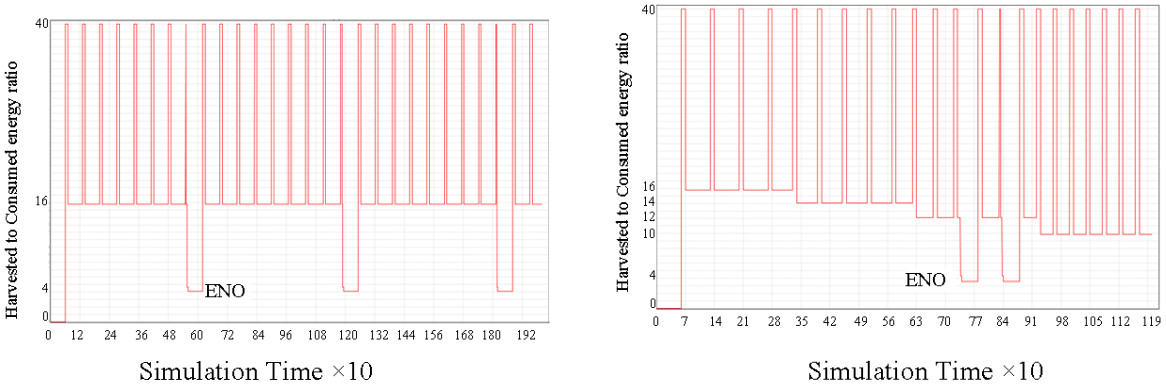
Figure 4.8: Consumed Energy for different sensing periods.

4.7.1.2 Energy-Neutral Operation State

The ratio HCR of the harvested energy over the consumed energy gives the ENO state of the node. Whenever $HCR > 1$, the node operates at a sustainable state. Figure 4.9.a depicts the operating state of the sensor ($simulate1[<= 2000]sender0.HCR$).

4.7.2 Dynamic Duty Cycle

In the second evaluation, the adjustment step is activated. The objective now is to achieve the ENO-MAX state by adjusting sensing period and beacon period. We set $SProb$ to 0.75 indicating that the sensing is the most important performance factor. Figure 4.9.b shows the optimization of HCR.



(a) ENO in static duty cycle (b) ENO in dynamic duty cycle

Figure 4.9: Operating state of the sender

After setting parameters, the system comes to stabilize. Figure 4.10 shows that the sensor arrives to the ENO-MAX where $HCR = 1$.

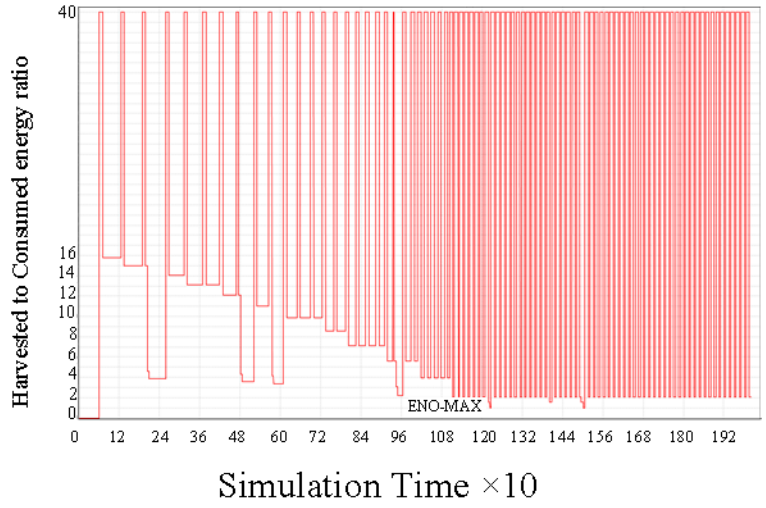


Figure 4.10: ENO-MAX state

The closer the HCR is to 1, the higher the sensing rate is (as shown in Figure 4.11).

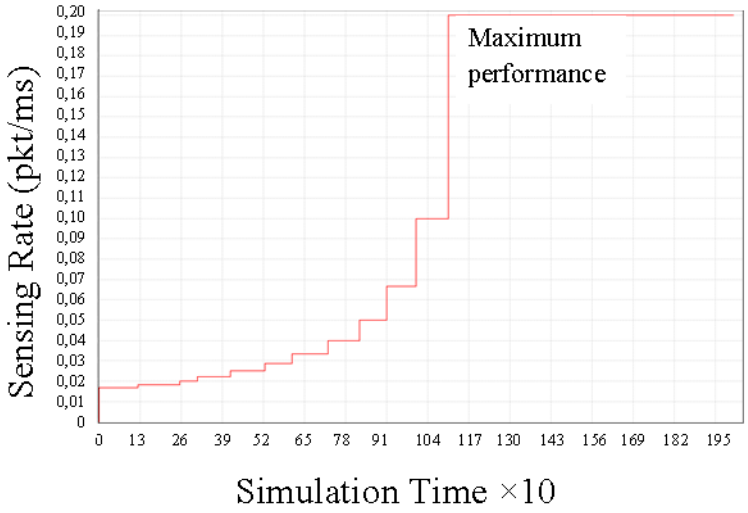


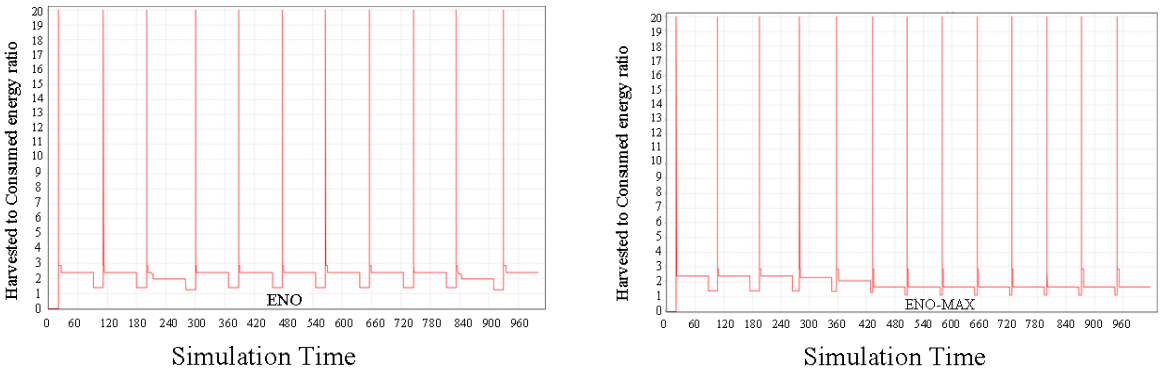
Figure 4.11: Sensing rate

Note that for both evaluations, the initial battery level is set to the sufficient battery level required to make at least one transmission. HCR is computed from the harvested and consumed energy at one cycle.

For the adjustment of the cycle, we first measure HCR. If it is above a certain threshold ($threshold > 1$), i.e., the consumed energy is less than the harvested energy, hence the node has more energy that it can exploit to optimize its performance. This optimization depends on $SProb$ value. If $SProb = 0.75$ (the throughput is the most important), then the sensing period is reduced each time until it reaches a minimum period. On the other hand, if the HCR is lower than the threshold, i.e., the node has a lack of energy, then it keeps the same sensing period to maintain the rate and increments the beacon period to conserve the energy. When

$SProb = 0.25$ (the delay is the most important), if the node has a surplus of energy, then it decreases the beacon period to increase the delay. Otherwise it increases the sensing period to reserve the energy. The last case is when $SProb = 0.5$ which mean that the throughput and the delay are equally important. Thus, the node decreases the sensing and the beacon periods to optimize the performance if it has enough energy and increases the periods otherwise.

Now, we take $SProb = 0.25$. The receiver decreases its beacon period and, eventually, decreases the delay and increases the power consumption. To see the optimization of HCR (HCR approaches 1), we first give the operating state of the node when the adjustment step is deactivated (Figure 4.12.a) then its ENO-MAX state in the case where the adjustment step is activated (Figure 4.12.b). In addition, Figure 4.13 shows the degradation of the delay in the second case.



(a) ENO in static duty cycle (b) ENO in dynamic duty cycle

Figure 4.12: Operating state of the receiver

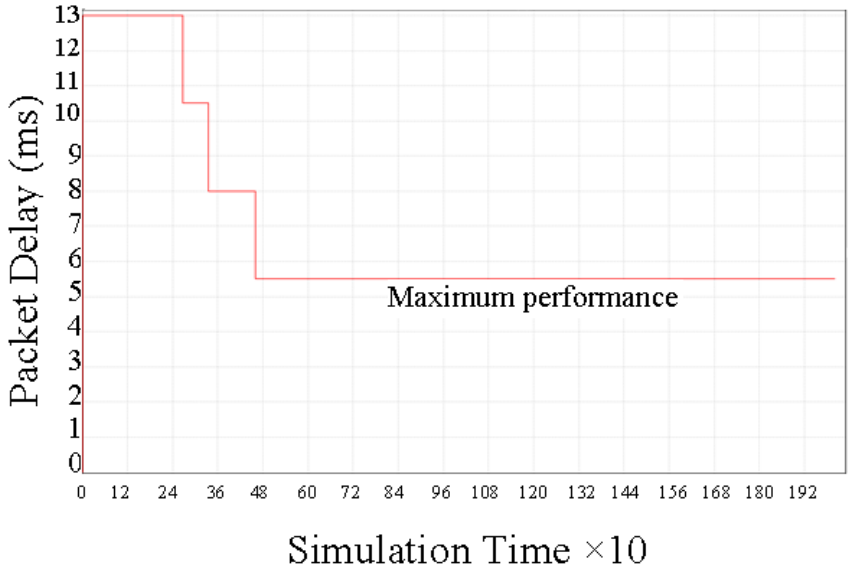


Figure 4.13: Packet delay

4.7.3 Variable Harvesting Rate

In all previous experiments, the harvesting rate is fixed which is not the case in reality. The energy harvesting rate is not constant and does not change regularly, but it can be influenced by many environmental and even human factors. Therefore the objective of this evaluation is to show the ability of the system to optimize its performance (delay or sensing rate) by exploiting or conserving its energy according to the available ambient energy. In order to reach that, the energy harvesting rate is fixed with $Mean = 1mA$ and $Variance = 0.2$. Moreover, the initial level of the battery is chosen to be at an optimal level and periodically is increased by the harvested amount (see Figure 4.6).

Under these conditions, we activate the adjustment step and take $SProb = 0.25$. Figure 4.14 depicts the consumed energy, the harvested energy and HCR. At each period $harv\_time$, the amount of the harvested energy is changed. We notice that the sender tries sending at the beginning when the energy is sufficient to consume more and consequently the HCR degrades in order to reach 1. Then, when the energy is decreased, the node reduces its consumption and tries to increase the HCR. Of course the amount of consumed energy depends on the activity done during this period (listening or sending).

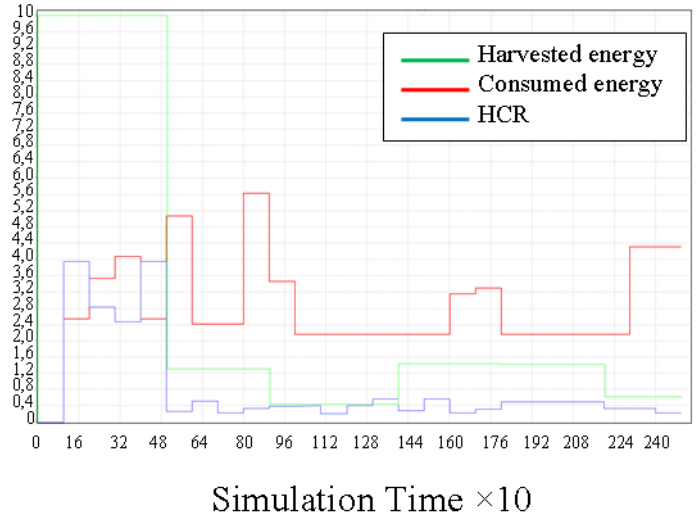
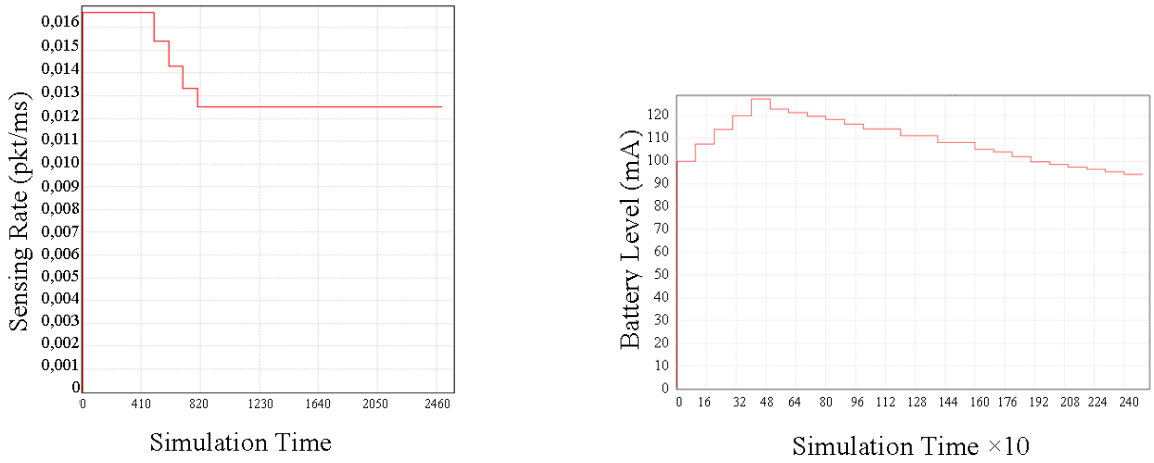


Figure 4.14: HCR for various energy harvesting rates.

As a result, the $sensing\_rate$ is decreased each time because the delay is the most important in this case (as shown in Figure 4.15.a). Eventually, the sensing rate stabilizes when the sensing period reaches its maximum. Moreover, Figure 4.15.b depicts the battery level.



(a) Sensing rate

(b) Battery level

Figure 4.15: Sensing rate and battery level for various energy harvesting rates

The last experiment shows the possibility that the node can be dead if it is far or hidden from the energy source. For that, we have to minimize the harvesting rate with $Mean = 0.1mA$ and $Variance = 0.02$ while the node cannot recover sufficient energy for communication. The query $Pr[<= 10000](<> sender0.Die)$ gives a probability interval $[0.9, 1]$ and Figure 4.16 shows the battery level.

Figure 4.16: Battery level with minimal harvesting rate.

4.8 Conclusion

This chapter has presented the modelling and formal verification of On Demand MAC (ODMAC) protocol [106], designed for Energy Harvesting WSNs (EH-WSNs). The protocol follows a receiver-initiated paradigm where a receiver which is available to receive data sends a beacon to all the other nodes. For the modelling, we have used stochastic timed automata (STA) which are suitable to model time constraints as well as probabilistic (or random) aspects of systems. We have proposed six STA models for the following components: receiver, sender, channel, adapter, capacitor, and harvester to represent the behaviour of each element intervenes in the EH-WSNs communication. For the analysis and the evaluation of the protocol, we have used the statistical model checking of the Uppaal tool. This evaluation has validated the principle of the protocol and also has showed the ability of a sensor to optimize its performance (sensing rate or packet delay, according to the application requirements) by conserving or consuming the available energy (depends on the harvesting rate) while keeping its ENO-MAX state as much as possible.

The formal verification of ODMAC, presented in this chapter, confirms the findings of previous evaluations in simulation [94] and in testbed-based experiments [97].

The next chapter focuses on another aspect of WSN which is mobility where a proposal for a new MAC protocol that combines mobility with energy harvesting will be detailed.

Chapter 5

A new Mobility and Energy Harvesting aware Medium Access Control (MEH-MAC) Protocol: Modelling and Performance Evaluation

5.1 Introduction

Existing medium access control (MAC) protocols have always considered the energy-harvesting aspect and mobility aspect separately. Incorporating mobility and energy harvesting into the same framework seems to be an important idea, but it poses new challenges for WSNs. Indeed, WSNs must improve their performances in a variety of scenarios. Mobility management requirement adds another dimension to WSNs protocols, in particular to the medium access control sub-layer protocols that are primarily responsible for scheduling, transmission and collision avoidance. In addition, energy harvesting frees the sensor node from the energy-restricted problem and places it before another type of challenge, which is to maintain itself in an energy-neutral operation state. Since wireless radio is the most energy-consuming component of the sensor, MAC protocols are also responsible in this case. In this chapter, a new "Mobility and Energy Harvesting aware Medium Access Control (MEH-MAC)" protocol [107] is proposed for dynamic sensor networks powered by ambient energy. It provides a detailed description of the protocol and presents a state-transition formal model with a performance evaluation using UPPAAL-SMC.

5.2 Related Work

Taking mobility into account in WSNs poses new challenges as the sensor nodes can frequently change locations and thus be deployed in various scenarios and cause fast topology changes. In this paper, we are interested in the challenges of mobility in protocols designing, in particular MAC protocols. Previous MAC protocols designed for WSNs focused primarily on improving energy efficiency and prolonging network lifetime. Therefore, managing mobility in this low energy condition motivates researchers to elaborate a number of contributions. In recent years, several surveys on mobility in WSNs have been carried out [108, 109, 110]. Among medium access protocols, sampling protocols (asynchronous protocols) reflect better the dynamics of mobile WSNs. Therefore, we are interested in works that promote asynchronous protocols to support mobility. Machiavel [111] is a B-MAC [57] extension more suitable for mobile sensor networks. In B-MAC, a sensor sends a preamble followed by a synchronization message before sending data. To handle mobility, Machiavel introduces a delay time between these last two operations of a static node or a mobile node can exploit it in order to occupy the channel. In this way, the mobile node allocates the channel already occupied by the preamble transmitter. X-Machiavel [112] expands Machiavel by applying the principles of X-MAC [96]. If the medium is occupied by a sensor which sends short preambles, the mobile node benefits from the interval between two consecutive preambles. Therefore, the mobile node sends its data to the preamble transmitter without having to precede its transmission with preambles. M-ContikiMAC [113] is an other sampling protocol. It has been proposed to overcome the shortcomings of the ContikiMAC protocol [114] when applied in a sensor network with mobile nodes. When constructing the routing table, mobile nodes are absent. When a mobile arrives in an area, it is not yet integrated into the routing tree and does not know its next destination. In order to avoid this complex and expensive management of the routing table, M-ContikiMAC proposes to free mobile nodes from multicast and unicast broadcasting, and limits its transmissions to anycast sending. MoX-MAC [115] is an improved version of X-MAC. As in X-MAC, a static transmitter continues to send short preambles until it receives an ACK frame from a static receiver. When a mobile node has data to send and the channel is busy, the mobile node senses the medium for ACK frame. Afterwards, it waits for the end of the ongoing transmission, then

it sends its data to the ACK receiver.

In addition to this mobility approach, the research community has integrated the energy harvesting field into WSNs in order to harness ambient energy to improve network performance. The challenge of energy harvesting approach is no longer energy but rather the conditions of recovery: the energy harvesting rate, the capacity of storage devices and the charging period. Under these conditions, the node must always ensure its operation in a sustainable energy state, i.e. it must guarantee that the consumed energy is always lesser than the harvested energy. Synchronous MAC protocols designed for ordinary WSNs maintain a common duty cycle due to the already recognized amount of energy. In contrast, in EH-WSN (Energy Harvesting-Wireless Sensor Networks), the amount of the available energy is not known in advance due to energy harvesting conditions. Therefore, asynchronous protocols serve well the needs of the EH-WSN where a sensor has its own cycle. This mechanism allows the low energy sensor to direct its cycle to minimize energy consumption independently of other nodes. In this regard, few MAC protocols have been proposed to support the proper functioning of nodes [77, 116]. Multi-Tier probabilistic polling (MTPP) [80] extends the single-hop probabilistic polling protocol [78] to perform multi-hop transmission. The nodes are assigned to tiers according to their distance from the sink which is represented by the number of hops. The sink broadcasts a polling message to all nodes of the first tier. The polled node broadcasts the same message again to higher level sensors. The procedure is repeated until all tiers are covered, the node elected from the last tier then begins the transmission of its packet. In order to manage the available energy, the sink adds a contention probability into the polling packet. Therfore nodes with insufficient energy are outside the competing sounding. In ODMAC [94], receivers initiate transmission by sending periodic beacons to senders indicating their availability. To acheive the ENO-MAX state (ENO state with maximum performance), a node adjusts the duty cycle by increasing or decreasing either the beacon period or the sensing period based on the energy harvesting rate and application requirements. QoS-Aware Energy-Efficient (QAEE) [117] protocol attaches importance to urgent data. Thus, the transmitter emits a beacon indicating the degree of urgency of its data. Therefore, the receiver wakes up earlier to collect all such beacons. Then, it responds with a beacon broadcast containing the highest priority node, allowing it to transmit while all other nodes go to sleep for the duration of that transmission. Additionally, nodes take their energy level into consideration when planning their duty cycles. A more recent protocol is DeepSleep [118]. DeepSleep is an energy harvesting based IEEE802.11 protocol specifically designed for machine to machine communication (M2M). In this mechanism, all nodes turn off their radios, except for the two communicating nodes. Deficient energy nodes also convert to Deep Sleep mode to harvest enough energy. Other MAC protocols designed for EH-WSN can be cited such as; Radio Frequency based Adaptive, Active Sleeping Period (RF-AASP) protocol [119], Adaptive Hierarchical MAC (AH-MAC) protocol [120], Energy-harvested Receiver-Initiated MAC (ERI-MAC) protocol [121] and Hybrid Asynchronous and Synchronous MAC protocol (HAS-MAC) [122].

5.3 MEH-MAC Description

In this section, we describe the basic communication in MEH-MAC. The idea behind our solution is to consider both aspects mobility and energy harvesting in the same framework. The nodes are powered by ambient energy and there are one or more nodes able to move around the network. The challenge in this solution is to be aware of two major problems. The first one, nodes are recovering energy from the environment with different rates almost minimal. Therefore MEH-MAC should control the consumed energy to prevent nodes from being turned

off due to power exhaustion. Hence, all nodes must be operating in the ENO state. The second, mobility in network causes frequent changes in topology. This can influence routing, access to the medium and energy consumption. When a mobile node arrives to a certain area with data that it detects, MEH-MAC must ensure that this data will be sent to nodes which forward it to the sink before the mobile node leaves that area. Likewise when a mobile node wishes to receive data (this type of mobility is used to collect data from the network), MEH-MAC must guarantee that the mobile node receives the data before its next movement.

5.3.1 Static Communication

MEH-MAC is an asynchronous protocol for duty cycling sensors that follows the receiver initiated paradigm [123]. Duty cycling mechanism is used in WSNs to reduce energy consumption by periodically putting off the node's radio. Therefore, each node in the network has tow periods, sleeping period and active period. In asynchronous protocols unlike synchronous one, nodes do not have a common schedule, but each has its own schedule. Hence nodes can wake up at different times. We prefer the asynchronous scheme for two reasons regarding energy harvesting and mobility. The first, a node in a charging state can not wake up at the scheduled time. The second, a mobile node can appear out of the schedule. Moreover, asynchronous protocols are further divided into sender initiated (so called, preamble sampling) and receiver initiated protocols. With the last paradigm when a receiver wakes up, it broadcasts a beacon to announce its availability to receive data from neighbours. Then it listens to the channel for incoming data for a period of time. A sender in active period waits until the reception of a beacon. After the reception of the appropriate beacon, the sender begins transmitting data. When the transmission ends, the sender and receiver both go into sleeping state.

Figure 5.1 below demonstrates the basic MEH-MAC communication between static receiver and static sender.

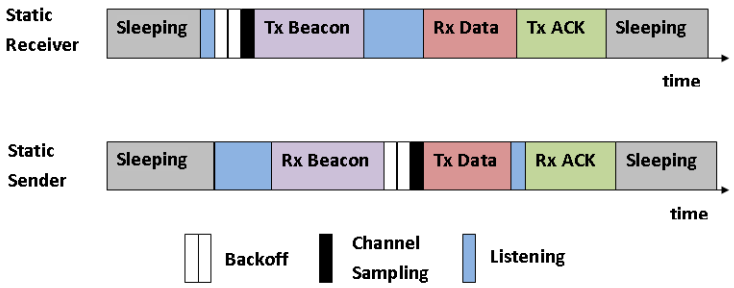


Figure 5.1: Basic communication between static receiver and static sender.

When a receiver enters an active period, it senses the channel for a time interval BT (Beacon Time). This sensing time is used to avoid the interruption of a transmission already declared by another receiver. If the channel remains busy after this time, the node returns to the sleeping state. If there is more than one receiver waking up at the same time, they must back off a random number of time slots. The winner will transmit his beacon and continue to wait for the data packet. If no data is received during DT (Data Time), the receiver goes into sleeping state. When the sender receives the beacon, it backs off a random time to minimize collision between senders waiting for the same beacon. This transmission should be acknowledged by the receiver. After a successful transmission, the sender and the receiver return to the sleeping state.

The following algorithms summarize the behaviour of a static sender and a static receiver. Each algorithm takes linear time with complexity of $O(n)$.

Algorithm 4 Static receiver

```
s:
time ← 0
while time ≤ forwarding_period do
    time ← time + T_slot
end while
listen_channel_BT()
if channel_free() then
    send_beacon()
    time ← 0
    while time ≤ DT ∧ ¬receive_data() do
        time ← time + T_slot
    end while
    if receive_data() then
        send_Ack()
    end if
    go to s                                          ▷ go to sleep
end if
```

Algorithm 5 Static sender

```
s:
time ← 0
while time ≤ sensing_period do
    time ← time + T_slot
end while
l:
time ← 0
while time ≤ waiting_time ∧ ¬receive_beacon() do
    time ← time + T_slot
end while
if ¬receive_beacon() then
    go to s                                          ▷ go to sleep
else
    BF ← T_slot * int[0, CW − 1]          ▷ delays a random number of time units
    listen_channel_BF()
    if channel_free() then
        send_data()
        receive_Ack()
        go to s                                      ▷ go to sleep
    else
        go to l                                      ▷ go to listening
    end if
end if
```

5.3.2 Dynamic Communication

Whenever a mobile node wakes up, it needs to send or collect data before the next movement. Therefore, MEH-MAC favours mobile nodes over static nodes when sending or receiving. Four cases are to be considered as follows.

1. *Mobile sender on a free channel:* If the mobile has data to sent, it first listens to the channel. If the channel is free, it waits for the reception of a beacon and immediately starts its transmission. The mobile node does not return to sleeping state unless its queue is empty. Figure 5.2 illustrates this situation.

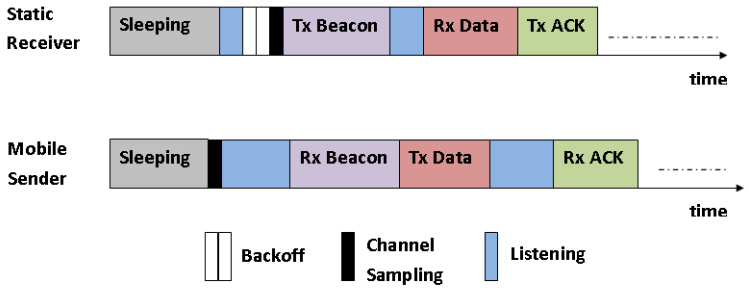


Figure 5.2: Communication with mobile sender when the channel is free.

2. *Mobile sender on a busy channel:* However, if the channel is busy, the protocol distinguishes two cases. The first when the channel is occupied by a beacon and the second when it is occupied by a data packet. In the former case, MEH-MAC allows the mobile node to possess the channel by preferring it over other senders. Figure 5.3 depicts this behaviour. Contrariwise, in the last case, the mobile node waits the end of the current transmission and broadcasts immediately a tiny frame Mob indicating its wish to sending data. Consequently, the receiver keeps its radio on and the senders waiting this receiver switch off their radios and go to sleeping. Figure 5.4 presents the communication between a static receiver, a static sender and a mobile sender.

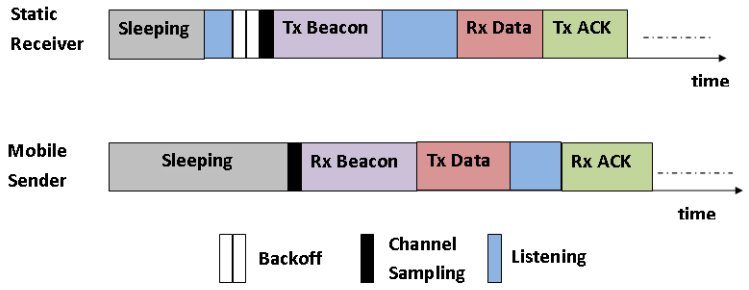


Figure 5.3: Communication with mobile sender when the channel is occupied with a beacon.

3. *Mobile receiver on a free channel:* When the mobile node wishes to receive data, it samples the channel. If this latter is free, it broadcasts a beacon without waiting. After the reception of the data packet, the mobile node retransmits a new beacon to collect more data before its next moving as shown in Figure 5.5.

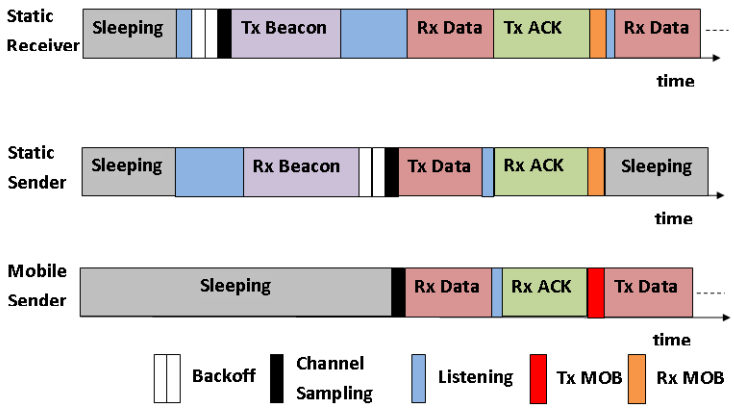


Figure 5.4: Communication with mobile sender when the channel is occupied with data.

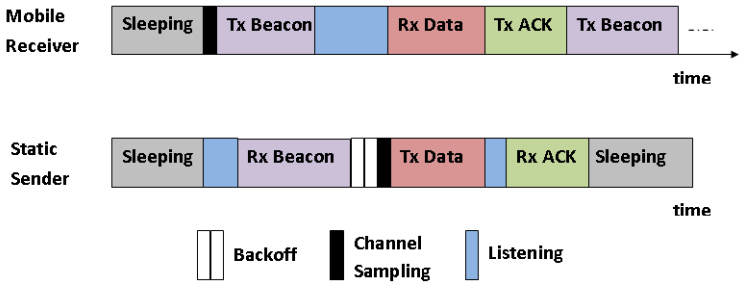


Figure 5.5: Communication with mobile receiver when the channel is free.

4. *Mobile receiver on a busy channel:* Now, if the channel is detected occupied, there are two cases mentioned hereafter. The first case when the channel is occupied with a beacon. The mobile node aborts the current beacon and sends its beacon as depicted in Figure 5.6. In the second case, the mobile node detects a data transmission. It waits for the end of this transmission and immediately broadcasts a beacon. The receiver goes to the sleeping state and the sender remains active in order to communicate with the mobile node (Figure 5.7).

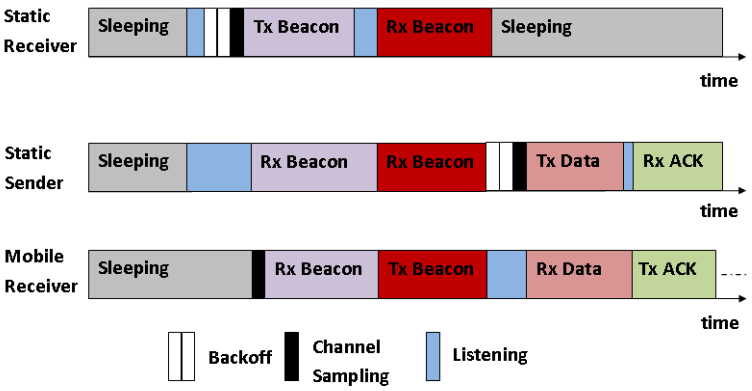


Figure 5.6: Communication with mobile receiver when the channel is occupied with a beacon.

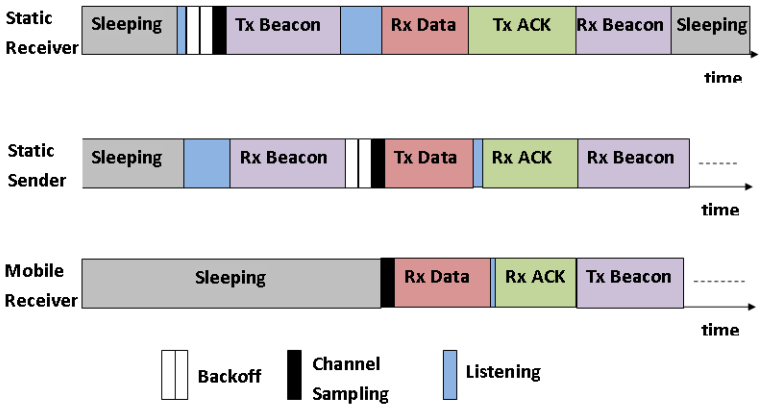


Figure 5.7: Communication with mobile receiver when the channel is occupied with a data packet.

The algorithms bellow summarize the communication with a mobile node. They have a complexity of $O(n)$.

Algorithm 6 Mobile sender

```
s:
while ¬wakeup() do                                              ▷ sleeping
end while
time ← 0
while time ≤ waiting_time ∧ ¬receive_beacon() do
    time ← time + T_slot
end while
if ¬receive_beacon() then
    go to s                                                     ▷ go to sleep
else
    if channel_free() then
r:
        send_data()
        receive_Ack()
        if more_queued_packets() then
            go to r                                             ▷ go to sending
        else
            go to s                                             ▷ go to sleep
        end if
    else
        if receive_beacon() then
            go to r                                             ▷ go to sending
        else
            if receive_data() then
                receive_Ack()
                send_Mob()
                go to r                                         ▷ go to sending
            end if
        end if
    end if
end if
```

Algorithm 7 Mobile receiver

```
s:
while ¬wakeup() do
                                                          ▷ sleeping
end while
if channel_free() then
b:
    send_beacon()
    time ← 0
    while time ≤ DT ∧ ¬receive_data() do
        time ← time + T_slot
    end while
    if ¬receive_data() then
        go to s                                           ▷ go to sleep
    else
        send_Ack()
        go to b                                           ▷ retransmit a new beacon
    end if
else
    if receive_beacon() then
        go to b                                           ▷ transmit a beacon
    else
        if receive_data() then
            receive_Ack()
            go to b                                       ▷ transmit a beacon
        end if
    end if
end if
```

5.3.3 Energy-neutral Operation State

In the two previous sections, we have detailed the behaviour of MEH-MAC when mobile nodes visit the network. In this section, we will focus on the harvest aspect and how nodes, whether static or dynamic, remain in the ENO state for as long as possible.

For the static nodes, we will adopt a mechanism similar to that used in [94]. Each static node has two duty cycles, one for forwarding and one for sensing. In order to ensure that the node is always in a sustainable state (ENO), it is necessary that the amount of energy consumed is always lesser than that harvested. For this reason, the period of each duty cycle can be adjusted to improve system performance when the node has excess energy or to conserve energy when the node has insufficient energy. The forwarding period controls the trade-off between delay and energy consumption. When the forwarding period is long, few beacons will be sent and little power will be consumed but the delay increases. However, when this period is small, a lot of beacons will be sent and therefore a lot of energy will be consumed with minimal delay. Likewise, the sensing period controls the trade-off between throughput and power consumption.

The objective of incorporating mobile sensors into WSNs depends on the application. The sensor can be placed on a person or animal to retrieve vital or behavioural informations. Additionally, it can be placed on a robot or a vehicle to monitor a specific environment. Mobile sensors can also move closer to the sink to reduce the power of data transmission. In most

of these applications, the mobile nodes should be awake when in proximity to static nodes to receive or send collected data. In a dense network, the mobile node must be awake most of the time. Therefore, the active period will be longer than the sleeping period. In order to meet these needs, static and dynamic nodes positions must be known at all times. Static node positions are fixed when the network is deployed. The positions of mobile nodes can be retrieved by a GPS card or RSSI (Received Signal Strength Indicator). Therefore, the mobile nodes turn off their radios in order to conserve energy and when approaching the static nodes, they turn on the radios to start communication with its neighbours. When a mobile node enters its active period, it does not return to its sleeping period unless it leaves its neighbours or has no packets to send and has not received any packets for a certain period of time. However, the consumed energy must always be less than the harvested energy in order to keep the mobile node operating in an energy neutral state. If the ratio of the harvested energy over the consumed energy falls below a certain threshold, the mobile node enters a charging state.

By following this process, a mobile node does not adopt a procedure for adjusting its periods as is the case with static nodes. If the mobile node consumes more energy in the case of dense networks or with hight traffic, then the node must harvest more energy in order to meet these constraints. A stable energy source will help solve this problem. These conditions lead us to think of vibration as a source of recoverable energy. With this idea, mobility takes advantage of energy harvesting to recover energy and energy harvesting takes advantage of mobility to generate energy. The design of an electric generator from the surrounding vibrations is therefore necessary. The mechanical energy of the vibration is able to provide enough energy to operate the mobile sensor. There are two types of transducers: electromagnetic and piezoelectric. In this work, we consider the first type based on animal body motion. Several works [124, 125, 126, 127] were competing to develop a generator that converts available energy as efficiently as possible. Considering that a generator is worn on a person's ankle to recover energy from their movements, then the mobile node can receive a sufficient amount of energy for several hours. The study that was carried out in [127] showed that the average power recovered when running is 649 μW and 123 μW when walking.

5.3.4 Doppler Effect Formulation

The moving of sensors in mobile WSN can cause frame drops due to the Doppler effect. When the sender changes its position, the signal frequency will be shifted which leads to bit errors in the frame. The resulting frequency shift is related to the position and relative velocity of the sender and the receiver. Doppler shift is defined as

$$f_d = \frac{v f_c}{c} \tag{5.1}$$

Where v is the relative velocity in meters/second, f_c is the carrier frequency of the radio and c is the speed of light.

In order to calulate the probability of successfully receiving a packet, one must first specify the radio propagation model. The log-normal shadowing model [128] is one of the most used models in WSNs. The following formula defined the model

$$PL(d) = PL(d_0) + 10nlog_{10}(\frac{d}{d_0}) + X_\sigma \tag{5.2}$$

Where $PL(d)$ is the path loss expressed in decibels related to the distance d between the sender and the receiver, d_0 is a reference distance, n is the path loss exponent and X_σ is Gaussian

random variable with mean equals zero and standard deviation σ which models the system losses.

The probability P of successfully receiving a frame when using non-return-to-zero (NRZ) coding is

$$P = (1 - P_e)^{8f} \tag{5.3}$$

Where f is the length of the frame in bytes and P_e is the bit error probability which depends on the modulation scheme and the Signal to Noise Ratio(SNR). P_e is defined by

$$P_e = \frac{1}{2} exp^{-\frac{\gamma}{2}\frac{B_N}{R}} \tag{5.4}$$

Where B_N represents the noise bandwidth in Hz, R is the data rate in bits and γ represents the SNR. The SNR in decibels relative to a distance d is defined in our case by

$$\gamma(d) = P_t - PL(d) - 10log_{10}f_d \tag{5.5}$$

where P_t is the power of transmission in decibels.

Therefore, $1 - P(d)$ is the probability of erroneous packet at the distance d, and if this probability is above a certain threshold then the packet is dropped.

5.3.5 Hand-off Handling

A hand-off issue occurs in WSN when the mobile node's link with the current receiver degrades. This event mainly happens in bursty traffic. The mobile node triggers a hand-off procedure when the quality of the link drops below a predefined threshold and stops it when it finds a new link with acceptable quality. To improve the proposed protocol, in this section, we will present a hand-off procedure to cover the mentioned problem.

We assume that not receiving acknowledgement is not sufficient to trigger a handover as it may be due to a collision rather than a weak signal. For this we will suggest the following mechanism. After sending a defined number of data packets, the mobile node initiates the procedure. It broadcasts tiny discovery packets and waits for a specified period until the reception of a limited number of acknowledgement packets. Upon reception, the mobile node chooses the node with the strongest signal and starts transmission again. Figure 5.8 summarises the hand-off procedure.

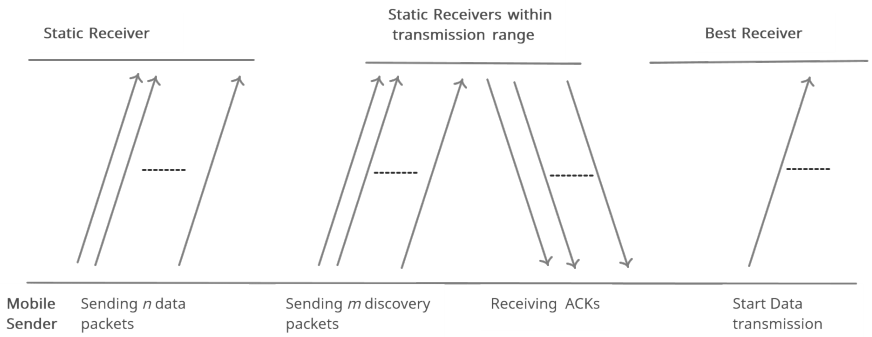


Figure 5.8: Hand-off procedure in MEH-MAC protocol.

We opt to use SNR (combined with Doppler shift) calculated in the previous section 5.3.4 as hand-off threshold. Therefore, the mobile node selects the sender of the acknowledgement packet with $SNR > threshold$. If there is more than one above the threshold, it selects the sender with the highest SNR.

5.4 MEH-MAC Modelling

After the detailed description of the protocol in the previous section, we now come to the formal verification. To do this, we have chosen the Uppaal tool. First of all, the protocol specification is modelled with stochastic timed automata. We have developed a model for each of the following behaviours. A static node can transmit (Figure 5.9), receive (Figure 5.10), adjust its cycle (Figure 5.11) and recover energy (Figure 5.12). A mobile node can move (Figure 5.15), transmit (Figure 5.13) and receive data (Figure 5.14). Both of nodes can monitor the state of their batteries (Figure 5.16). We have elaborated also a model to represent the channel that manages the synchronization between the different entities of the system. All the figures are available on the following link: `https://drive.google.com/drive/folders/1BprFL2xn-BhY02RIjZyEQQJDU9iigs6Q?usp=sharing`

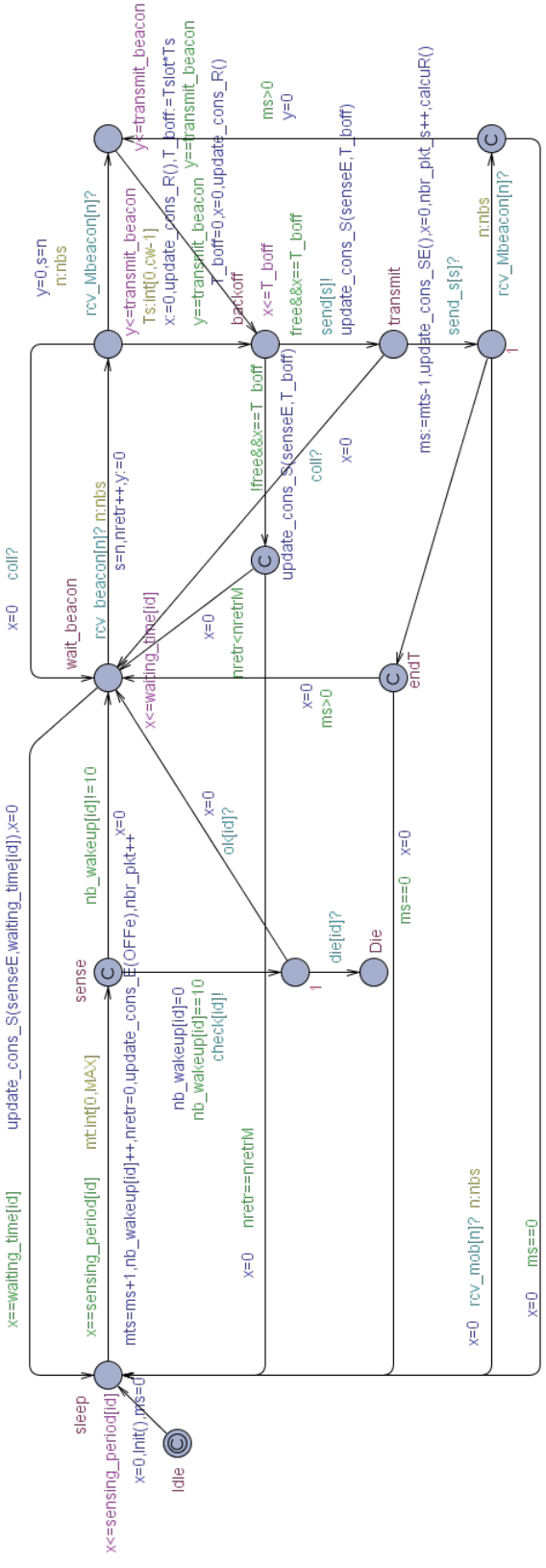


Figure 5.9: Stochastic timed automata of a static sender.

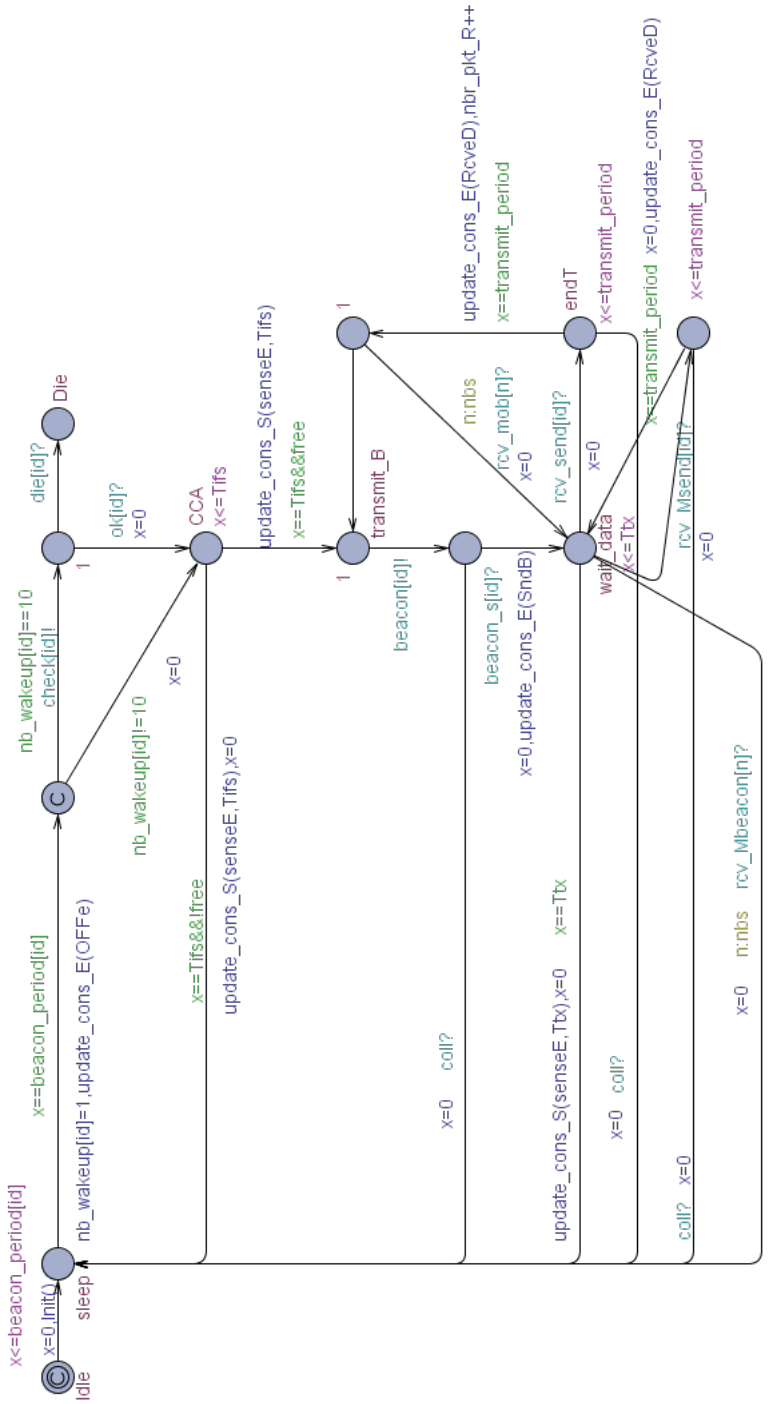


Figure 5.10: Stochastic timed automata of a static receiver.

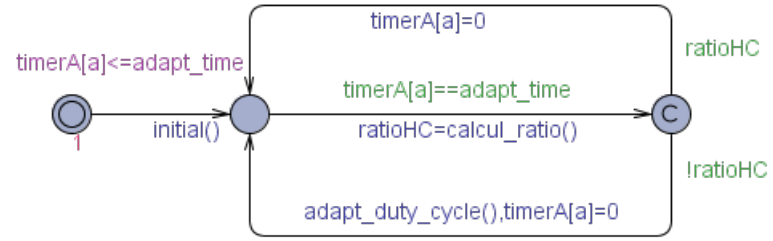


Figure 5.11: Stochastic timed automata of cycle adjustment.

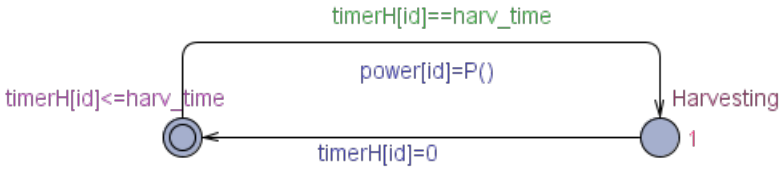


Figure 5.12: Stochastic timed automata of the harvester.

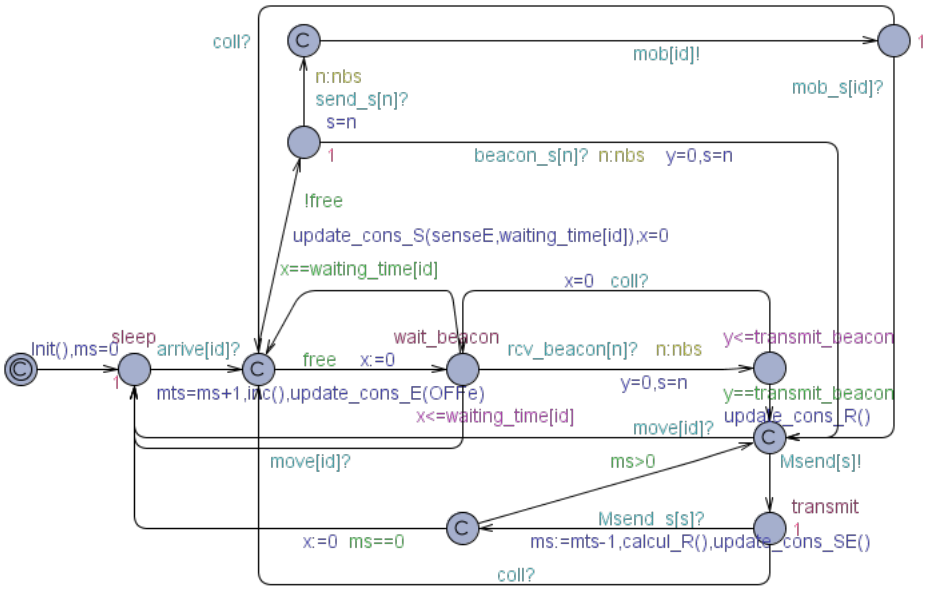


Figure 5.13: Stochastic timed automata of mobile sender.

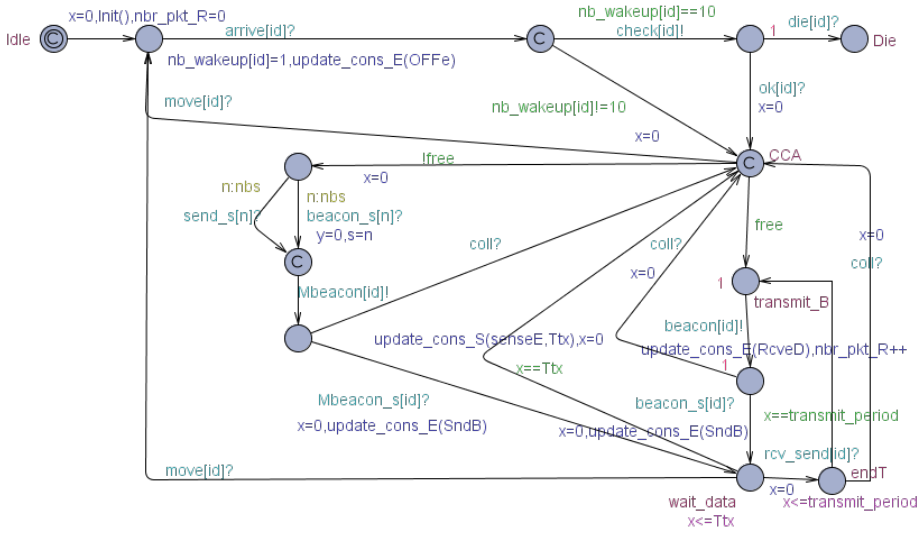


Figure 5.14: Stochastic timed automata of mobile receiver.

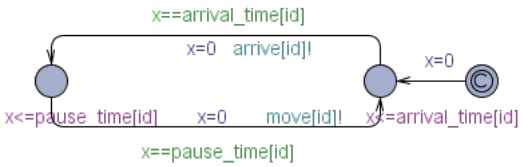


Figure 5.15: Stochastic timed automata of the clock.

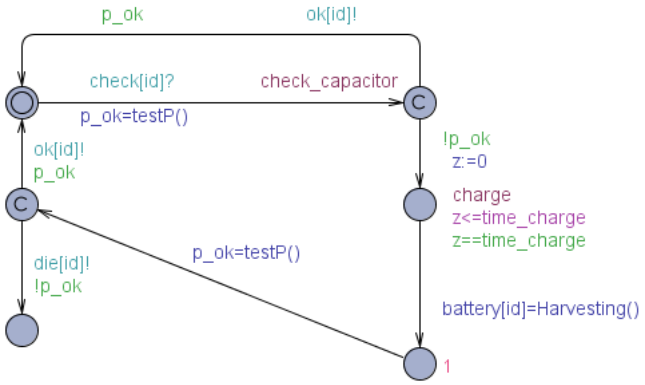


Figure 5.16: Stochastic timed automata of the battery checker.

5.5 Case Study for the Analysis Phase

The proposed protocol can be implemented in a WSN for animal tracking. Animals are detected either by static sensor nodes (SS) placed in the coverage area or by mobile sensor nodes (MS) installed on the animal's limbs (neck, leg, ankle, etc.). The information collected is then sent to the sink which plays the role of a gateway to send this data to a remote server via the internet or any other communications system for further future studies. In addition to

sensors, animals are equipped with generators to recover energy from their movements. For mobility, we opt for the Random Way point model where the mobile node randomly chooses its destination, speed of movement and a pause time as it reflects well the behaviour of animals. The architecture of this solution is depicted in Figure 5.17.

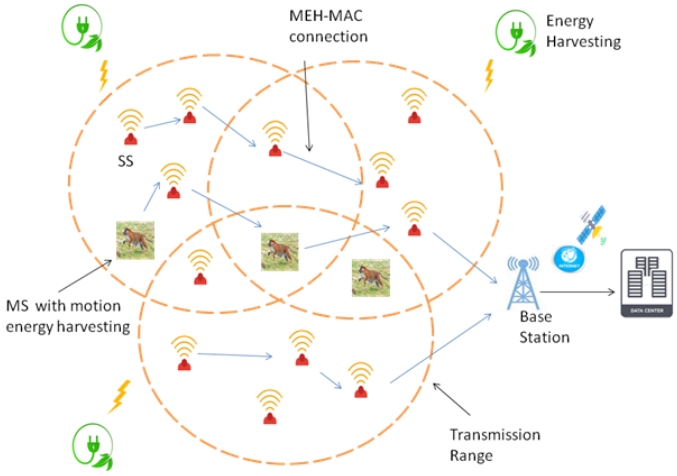


Figure 5.17: System Architecture.

5.6 Analysis and Evaluation

In this analysis, we will rely on a comparison between the behavior of ODMAC [94] and MEH-MAC when a mobile sensor visits the network. The mobile sensor has a pause time and a travel speed. It has a GPS card and knows the positions of all the other nodes. As ODMAC and MEH-MAC are asynchronous protocols, the sending of the packets is in any-cast. The performance criterion to be evaluated is the ratio of the number of sent packets to the total number of packets. This criterion is specified in PCTL language of the UPPAAL-SMC with the query: $simulate1[<=5000]MobileSender2.rpkt$.

The simulation parameters used in this evaluation are depicted in Table 5.1.

Table 5.1: Simulation parameters.

Parameter	Value
Data rate	256 $kbps$
Message length	100 $bytes$
Beacon size	8 $bytes$
Data packet transmission period	3.125 ms
Beacon packet transmission period	0.25 ms
DT	66 ms
T_{slot}	1 ms
CW	63
BT	70 ms
Charging_time	10 s
Initial energy	560 mA
Transmission power	10 dBm
Power consumed while transmitting	26.7 mA
Power consumed while receiving	9.2 mA
Power consumed while Idle	6 mA
Power consumed while sleep	1 μA
Mobility model	Random Way Point

5.6.1 ODMAC Evaluation

In the first evaluation, all sensors use ODMAC protocol to access the medium. Figure 5.18 shows that a static node always has priority or the same chance in best case of sending its packet. Therefore, the number of lost packets for a mobile node is always high (with a minimum ratio).

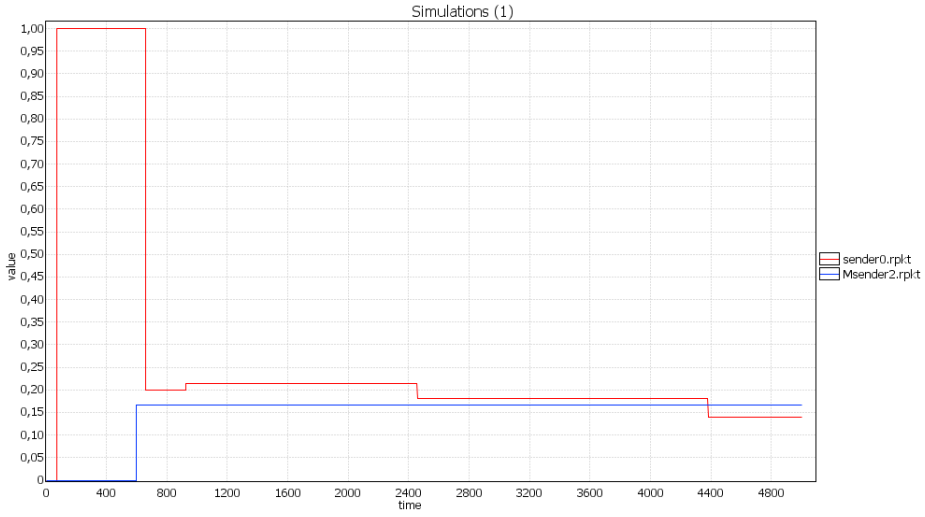


Figure 5.18: The ratio of the number of packets sent by a static and dynamic node to the total number of packets in ODMAC.

Likewise, Figure 5.19 shows that a static receiver node receives more packets than a mobile one.

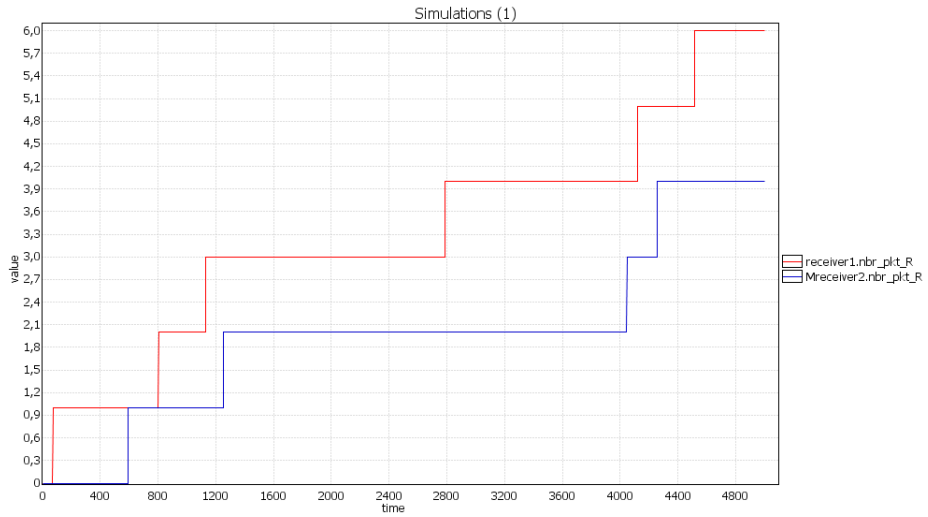


Figure 5.19: The total packets received by a static receiver and a mobile receiver in ODMAC.

5.6.2 MEH-MAC Evaluation

In the second evaluation, all sensors use MEH-MAC protocol to access the medium. Figure 5.20 shows that the mobile node has successfully transmitted its packets (with ratio=1). The reason is that in MEH-MAC protocol, mobile nodes always take priority.

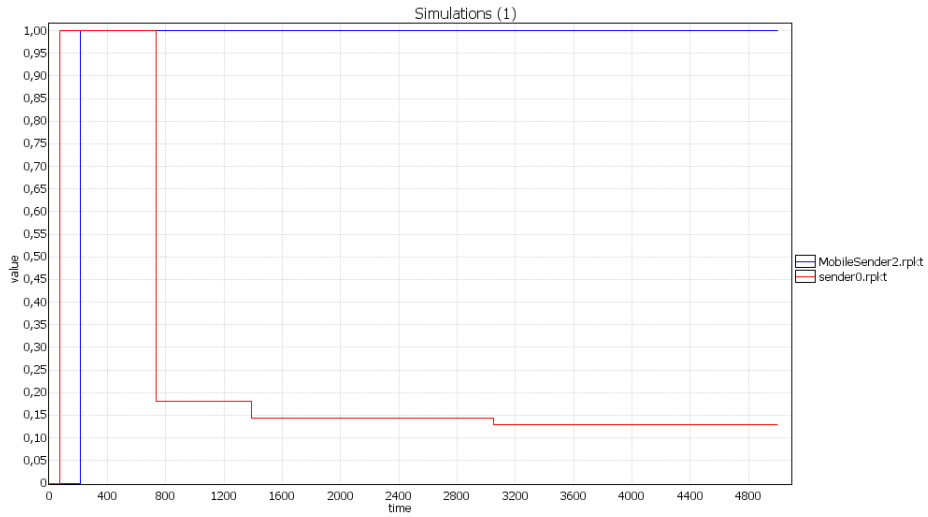


Figure 5.20: The ratio of the number of packets sent by a static and dynamic node to the total number of packets in MEH-MAC.

In Figure 5.21, the mobile receiver node receives more packets than the static node.

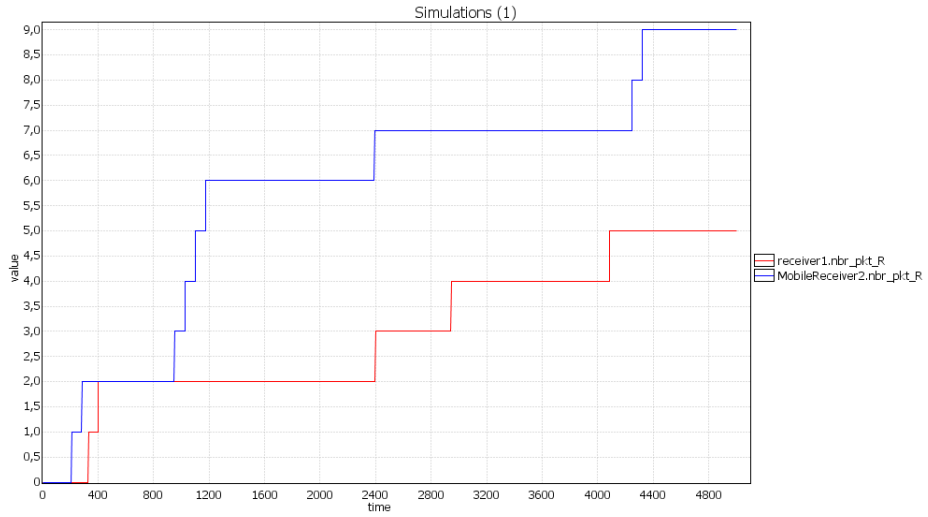


Figure 5.21: The total packets received by a static receiver and a mobile receiver in MEH-MAC.

5.6.3 ENO State Evaluation

An energy harvesting MAC protocol must ensure that all nodes are in their sustainable state where the consumed energy does not exceed the harvested energy. A node is in its ENO-MAX state when it reaches ENO state at maximum performance. The objective of this assessment is to demonstrate that MEH-MAC is able to guarantee that all nodes are in their ENO state.

A static node (sender or receiver) can adjust its cycle (sensing period or forwarding period) by reducing or increasing it depending on the amount of available energy and the performance criteria (sensing rate or delay) of the application.

The static nodes of the network reduce the sensing period in order to achieve the maximum sensing rate if this is the most important criterion. Figure 5.22 shows the increase of the sensing rate. On the other hand, if the delay is the most important, the nodes will reduce the forwarding period. Figure 5.23 shows the decrease of the delay. Note that the energy harvesting rate of static nodes is modelled as random variable that follows a normal distribution with $Mean = 1mA$ and $Variance = 0.2$.

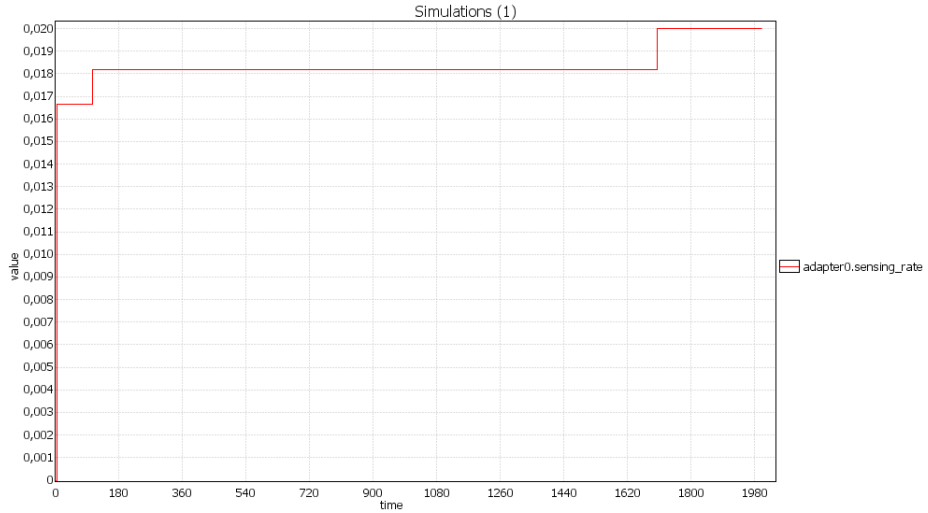


Figure 5.22: The increase of the sensing rate by adjustment of the cycle.

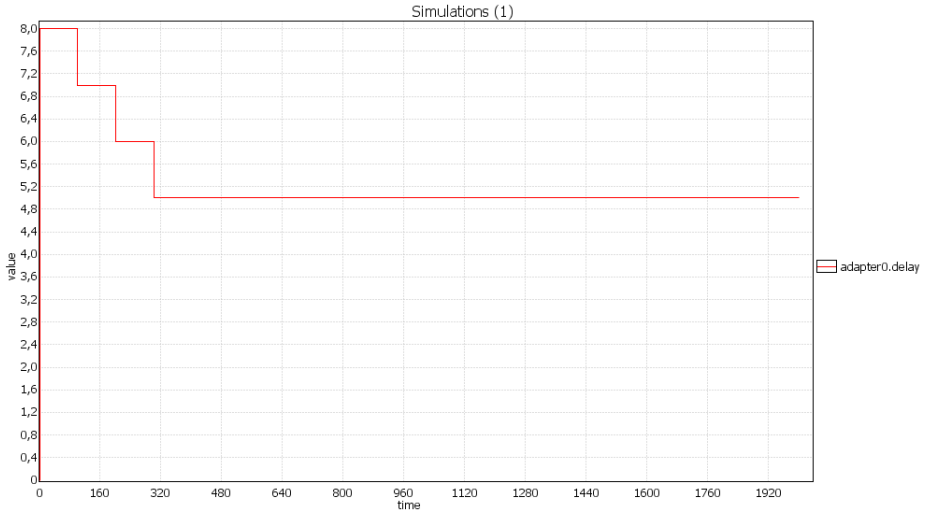


Figure 5.23: The decrease of the delay by adjustment of the cycle.

Figure 5.24 shows that the node mobile has succeeded in operating in its sustainable state.

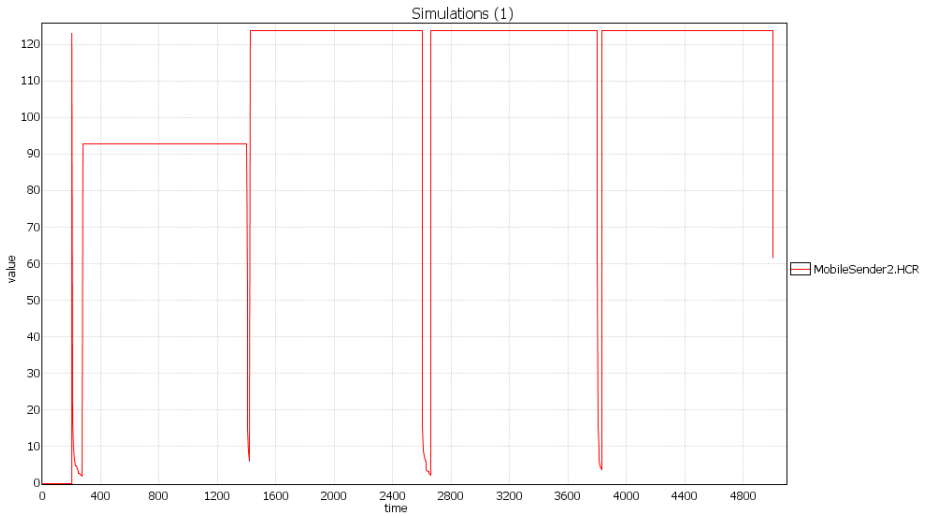


Figure 5.24: Energy-Neutral Operation (ENO) state of the mobile node.

5.6.4 Packet Error Evaluation

In this evaluation, we are interested in the number of packets dropped due to errors caused by frequency shift during the movement of mobile nodes. The speed of the mobile node and the time pause are chosen randomly by a uniform distribution over maximum values. The parameters used in this experiment are listed in Table 5.2 below. As we saw in Eq. 5.1, Doppler effect is correlated with the velocity of the mobile node, therefore, high speed results in a large shift. Figure 5.25 shows that the ratio between the number of dropped packets to the total number increases with increasing speed.

Table 5.2: Noise parameters.

Parameter	Value
f_c	$433.3MHz$
d_0	$1Km$
$PL(d_0)$	$55dB$
n	4
σ	4
B_N	$30KHz$

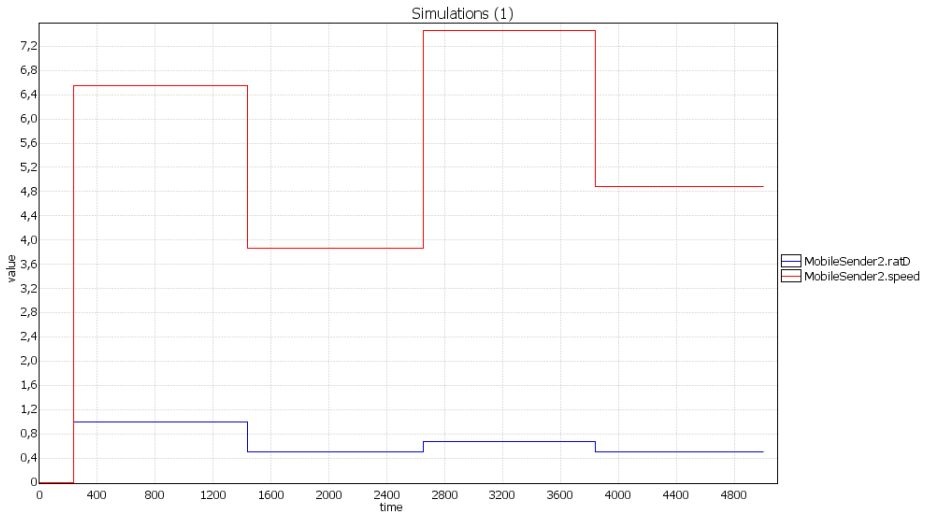


Figure 5.25: Impact of speed on the number of dropped packets.

5.6.5 Hand-off Evaluation

In order to evaluate the hand-off mechanism, we will make a comparison between MEH-MAC and its improved version with this mechanism. For this, we will calculate the number of lost packets of a mobile node with the two protocols. In the simulation, we assumed that at a certain moment the connection between the mobile node and its receiver is degraded due to its movement. Without the hand-off mechanism, the mobile node continues to send data packets to a receiver outside its transmission range which leads to the loss of these packets. On the other hand, with the mechanism, the mobile node stops the transmission, and it will start searching for a receiver with an acceptable signal, then it begins its transmission again.

Figure 5.26 proves the efficiency of the hand-off mechanism, as the mobile node successfully finds a receiver and the ratio of the number of sent packets to the total number of packets reaches 1.

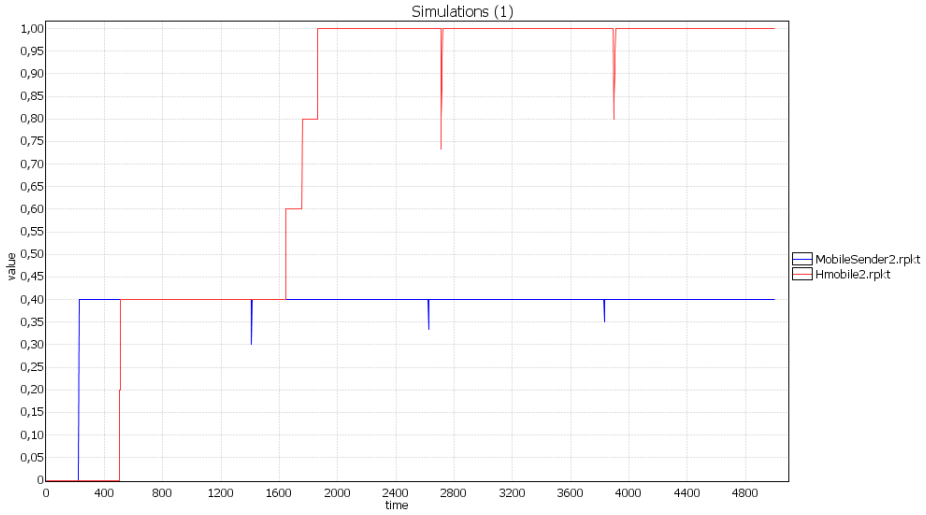


Figure 5.26: The ratio of the number of sent packets to the total number of packets with MEH-MAC (in blue) and its improved version (in red).

5.7 Comparison with Recent Works

The majority of MAC protocols developed for WSN do not integrate both mobility and harvesting aspects as described in previous sections (Section 5.1 and Section 5.2). MEH-MAC combines the two concepts in the same framework, managing the challenges posed by each of them. In this section, we will summarize the most recent works in the field in order to compare them with the proposed protocol. Table 5.3 below sums up the most important evaluation criteria for these works.

5.8 Conclusion

The proposed protocol in this paper is a MAC protocol designed for dynamic and energy harvesting sensor networks. Thus, the network consists of static and mobile nodes where the nodes are supplied with recoverable energy. For mobile nodes, a stable power supply is essential. For this, we proposed to take advantage of mobility in order to generate energy. As an example, we took a animal's movement condition to validate the model. Regarding access to the medium, the nodes follow an asynchronous policy. When a receiver is ready to receive packets, it sends a beacon to the transmitters. When a mobile node arrives, it has the priority in sending or receiving packets to ensure that it finishes its job before the next movement. On the other hand, in order to ensure that all nodes are operating in their sustainable states, a static node cycle adjustment mechanism similar to that proposed in ODMAC [94] is used. In order to validate the proposed protocol, modelling and probabilistic validation were established using UPPAAL-SMC. The obtained results confirm the specification of the protocol and validate its proposal. As a future work, we considered the implementation and application of the protocol in a real environment.

Table 5.3: Comparison with recent works.

	Energy harvesting	Mobility	Performance metrics	Medium Access	Evaluation method
[122]	✓	✗	End-to-End delay	Asynchronous and Synchronous	Simulation
[129]	✓	✗	Throughput, Packet loss rate	Asynchronous	Simulation
[130]	✓	✗	Throughput	Asynchronous	Simulation
[131]	✗	✓	Throughput, Energy consumption, Packet delay	Asynchronous and Synchronous	Simulation
[132]	✗	✓	Energy consumption, Packet latency, Packet delivery ratio Network sustainability,	Asynchronous	Simulation
[133]	✓	✓	Throughput, Packet loss, Packet delay	Synchronous	Simulation
MEH-MAC	✓	✓	Packet loss, ENO state, Doppler shift, Dropped packets, Packet delay, Sensing rate	Asynchronous	Formal verification

GENERAL CONCLUSION

Thesis summary

The main limitation of the sensor nodes used in WSNs is the limited battery capacity with the difficulty of changing them, so that several solutions have been proposed to maximize their lifetime. Energy saving in MAC layer is basically achieved by energy-aware MAC protocols through minimizing packet collisions, idle listening, overhearing and especially turning off the radio as much as possible. While these protocols help prolonging the sensor lifetime, this latter remains bounded and finite. Energy harvesting is an alternative approach that is being applied to extend WSN lifetime. Therefore, sensor nodes are able to convert harvested energy from the surrounding environment sources into electricity to power themselves. Since ambient energy availability is random and irregular in nature, proper MAC protocols are needed to accurately check the remaining power in nodes.

On the other hand, the introduction of mobility in WSNs offers multiple benefits such as extending the network coverage, improve the routing performances or the overall connectivity. However, it also poses several challenges such as frequent topological changes, intermittent connectivity, increase in collision rate and consequently an increase in energy consumption. Green computing and more specifically energy harvesting is a promising technology to solve this problem. By combining energy harvesting technologies with mobility, efficient computing performance can be achieved.

Efficient sensor data transmission depends on the correct behaviour of MAC protocols. Therefore, it is important to choose a technique to verify the behaviour of these protocols before its implementation. Some drawbacks can be discovered by simulations and testing. However, formal verification techniques such as model checking, which enumerates all possible paths and automatically checks specified properties, have become a suitable option for system validation. Formal verification has proved its usefulness in WSNs where many works use it to validate their proposals.

The contributions in this thesis revolve around the use of formal techniques in the development and evaluation of WSNs. The work carried out consists of:

- Proposing a stochastic model representing the 802.11 MAC protocol, and a probabilistic evaluation of the system performance.
- Modelling and formal verification of a MAC protocol designed for EH-WSNs networks using statistical model checking.
- Finally, the proposal and the formal evaluation of a new MAC protocol for dynamic sensor networks powered by ambient energy.

Future directions

In order to extend the work carried out in this thesis, we propose the following topics:

- Carry out an implementation of the proposed protocol and validate the model in systematic field experiments.
- Address other layers, especially the network layer, to improve the protocol.
- Propose optimizations to existing model-checking algorithms.

Bibliography

[1] Marta Kwiatkowska, Gethin Norman, and Jeremy Sproston. Probabilistic model checking of the ieee 802.11 wireless local area network protocol. In *Joint International Workshop von Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, pages 169–187. Springer, 2002.

[2] Matthias Fruth. Probabilistic model checking of contention resolution in the ieee 802.15. 4 low-rate wireless personal area network protocol. In *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (isola 2006)*, pages 290–297. IEEE, 2006.

[3] Matthias Fruth. *Formal methods for the analysis of wireless network protocols*. PhD thesis, University of Oxford, 2011.

[4] Paolo Ballarini, Hilal Djafri, Marie Duflot, Serge Haddad, and Nihal Pekergin. COSMOS: A statistical model checker for the hybrid automata stochastic logic. In *Proc. 8th International Conference on Quantitative Evaluation of SysTems (QEST'11)*, pages 143–144. IEEE CS Press, September 2011.

[5] Youcef Hammal, Jalel Ben-Othman, Lynda Mokdad, and Abdelkrim Abdelli. Formal modeling and verification of an enhanced variant of the ieee 802.11 csma/ca protocol. *Journal of Communications and Networks*, 16(4):385–396, 2014.

[6] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[7] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[8] Beom-Su Kim, HoSung Park, Kyong Hoon Kim, Daniel Godfrey, and Ki-Il Kim. A survey on real-time communications in wireless sensor networks. *Wireless Communications and Mobile Computing*, 2017, 2017.

[9] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of supercomputing*, 68(1):1–48, 2014.

[10] Javad Rezazadeh. Mobile wireles sensor networks overview. *International Journal of Computer Communications and Networks (IJCCN)*, 2(1), 2012.

[11] Silicon Labs. The evolution of wireless sensor networks. *International Journal of Computer Communications and Networks (IJCCN)*, 9, 2013.

[12] Chee-Yee Chong and Srikanta P Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[13] Th Arampatzis, John Lygeros, and Stamatis Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 719–724. IEEE, 2005.

[14] Aarti Kochhar, Pardeep Kaur, Preeti Singh, and Sukesha Sharma. Protocols for wireless sensor networks: a survey. *Journal of Telecommunications and Information Technology*, 2018.

[15] Calypso Barnes. *Verification and validation of wireless sensor network protocol properties through the system's emulation*. PhD thesis, Université Côte d'Azur, 2017.

[16] Edmund M Clarke and Jeannette M Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)*, 28(4):626–643, 1996.

[17] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[18] Jerry R Burch, Edmund M Clarke, Kenneth L McMillan, David L Dill, and Lain-Jinn Hwang. Symbolic model checking: 1020 states and beyond. *Information and computation*, 98(2):142–170, 1992.

[19] Kenneth L. McMillan. Symbolic model checking: an approach to the state explosion problem. 1992.

[20] Alessandro Cimatti, Edmund M. Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An open-source tool for symbolic model checking. In *Proceedings of the 14th International Conference on Computer Aided Verification*, CAV '02, pages 359–364, London, UK, UK, 2002. Springer-Verlag.

[21] Armin Biere, Alessandro Cimatti, Edmund M Clarke, Ofer Strichman, Yunshan Zhu, et al. Bounded model checking. *Advances in computers*, 58(11):117–148, 2003.

[22] Armin Biere, Alessandro Cimatti, Edmund Clarke, and Yunshan Zhu. Symbolic model checking without bdds. In *International conference on tools and algorithms for the construction and analysis of systems*, pages 193–207. Springer, 1999.

[23] Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking: An overview. In *International conference on runtime verification*, pages 122–135. Springer, 2010.

[24] Peter Bulychev, Alexandre David, Kim Gulstrand Larsen, Marius Mikučionis, Danny Bøgsted Poulsen, Axel Legay, and Zheng Wang. Uppaal-smc: Statistical model checking for priced timed automata. *arXiv preprint arXiv:1207.1272*, 2012.

[25] Musab AlTurki and José Meseguer. Pvesta: A parallel statistical model checking and quantitative analysis tool. In *International Conference on Algebra and Coalgebra in Computer Science*, pages 386–392. Springer, 2011.

[26] Junaid Qadir and Osman Hasan. Applying formal methods to networking: theory, techniques, and applications. *IEEE Communications Surveys & Tutorials*, 17(1):256–291, 2014.

[27] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*, volume 2050. Springer Science & Business Media, 2001.

[28] Jos CM Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2-3):131–146, 2005.

[29] Philip Merlin and David Farber. Recoverability of communication protocols-implications of a theoretical study. *IEEE transactions on Communications*, 24(9):1036–1043, 1976.

[30] Youcef Atamna. *Réseaux de Petri temporisés stochastiques classiques et bien formés. Définition, analyse et application aux systèmes distribués temps réel*. PhD thesis, Toulouse 3, 1994.

[31] Josephus CM Baeten, Twan Basten, Twan Basten, and MA Reniers. *Process algebra: equational theories of communicating processes*, volume 50. Cambridge university press, 2010.

[32] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

[33] Alexandre Mouradian. *Proposition et vérification formelle de protocoles de communications temps-réel pour les réseaux de capteurs sans fil*. PhD thesis, 2013.

[34] Marta Kwiatkowska, Gethin Norman, and Jeremy Sproston. Symbolic computation of maximal probabilisti reachability. In *International Conference on Concurrency Theory*, pages 169–183. Springer, 2001.

[35] Marta Kwiatkowska, Gethin Norman, David Parker, and Jeremy Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 29(1):33–78, 2006.

[36] Gerd Behrmann, Kim G Larsen, and Jacob I Rasmussen. Priced timed automata: Algorithms and applications. In *International Symposium on Formal Methods for Components and Objects*, pages 162–182. Springer, 2004.

[37] Kristin Y Rozier. Linear temporal logic symbolic model checking. *Computer Science Review*, 5(2):163–203, 2011.

[38] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57. IEEE, 1977.

[39] Edmund M. Clarke, E Allen Emerson, and A Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263, 1986.

[40] Rajeev Alur, Costas Courcoubetis, and David Dill. Model-checking in dense real-time. *Information and computation*, 104(1):2–34, 1993.

[41] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal aspects of computing*, 6(5):512–535, 1994.

[42] Andre Barroso, Utz Roedig, and Cormac Sreenan. /spl mu/-mac: an energy-efficient medium access control for wireless sensor networks. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.*, pages 70–80. IEEE, 2005.

[43] Matthew J Miller and Nitin H Vaidya. A mac protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Transactions on mobile Computing*, (3):228–242, 2005.

[44] Sandeep S Kulkarni. Tdma service for sensor networks. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, pages 604–609. IEEE, 2004.

[45] Ilker Demirkol, Cem Ersoy, and Fatih Alagoz. Mac protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.

[46] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1567–1576. IEEE, 2002.

[47] Amit Sinha and Anantha Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Design & Test of Computers*, (2):62–74, 2001.

[48] Raed Alsaqour AfiqahAzahari, Mohammed Al-Hubaishi, and Mueen Uddin. Review of error detection of data link layer in computer network. *Middle-East Journal of Scientific Research*, 18(7):968–973, 2013.

[49] James F Kurose and Keith W Ross. *Computer networking: a top-down approach*. Addison Wesley, 2011.

[50] Raghu K Ganti, Praveen Jayachandran, Haiyun Luo, and Tarek F Abdelzaher. Datalink streaming in wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 209–222. ACM, 2006.

[51] Hang Liu, Hairuo Ma, Magda El Zarki, and Sanjay Gupta. Error control schemes for networks: An overview. *Mobile networks and Applications*, 2(2):167–182, 1997.

[52] Henri Dubois-Ferrière, Deborah Estrin, and Martin Vetterli. Packet combining in sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 102–115. ACM, 2005.

[53] Allen Miu, Hari Balakrishnan, and Can Emre Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 16–30. ACM, 2005.

[54] Urmila A Patil, Smita V Modi, and BJ Suma. A survey: Mac layer protocol for wireless sensor networks. *International Journal of Emerging Technology and Advanced Engineering*, 3(9):203–211, 2013.

[55] T Rajendran. A survey on mac protocols for wireless sensor networks. In *Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering*, pages 121—-126. ACSIS, 2017.

[56] IEEE Computer Society LAN MAN Standards Committee et al. Wireless lan medium access control (mac) and physical layer (phy) specifications. *ANSI/IEEE Std. 802.11-1999*, 1999.

[57] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, 2004.

[58] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking (ToN)*, 12(3):493–506, 2004.

[59] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, 2003.

[60] Murat Dener and Ömer Faruk Bay. Medium access control protocols for wireless sensor networks: Literature survey. *Gazi University Journal of Science*, 25(2):455–464, 2012.

[61] Venkatesh Rajendran, Katia Obraczka, and Jose Joaquin Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless networks*, 12(1):63–78, 2006.

[62] Gang Lu, Bhaskar Krishnamachari, and Cauligi S Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, page 224. IEEE, 2004.

[63] Sungrae Cho, Kalyani Kanuri, Jin-Woong Cho, Jang-Yeon Lee, and S-D June. Dynamic energy efficient tdma-based mac protocol for wireless sensor networks. In *Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services-(icas-isns' 05)*, pages 48–48. IEEE, 2005.

[64] Anis Koubâa, Mário Alves, and Eduardo Tovar. Ieee 802.15. 4: a federating communication protocol for time-sensitive wireless sensor networks. *Sensor Networks and Configurations: Fundamentals, Techniques, Platforms, and Experiments*, pages 19–49, 2006.

[65] Nick Baker. Zigbee and bluetooth: Strengths and weaknesses for industrial applications. *Computing and Control Engineering*, 16(2):20–25, 2005.

[66] Amre El-Hoiydi. Spatial tdma and csma with preamble sampling for low power ad hoc wireless sensor networks. In *Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications*, pages 685–692. IEEE, 2002.

[67] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, and Mihail L Sichitiu. Z-mac: a hybrid mac for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(3):511–524, 2008.

[68] Jaehyun Kim, Jeongseok On, Seoggyu Kim, and Jaiyong Lee. Performance evaluation of synchronous and asynchronous mac protocols for wireless sensor networks. In *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, pages 500–506. IEEE, 2008.

[69] Bo Fu, Yang Xiao, Hongmei Julia Deng, and Hui Zeng. A survey of cross-layer designs in wireless networks. *IEEE Communications Surveys & Tutorials*, 16(1):110–126, 2013.

[70] Gianluigi Ferrari. *Sensor Networks: where theory meets practice*. Springer Science & Business Media, 2010.

[71] Islam T Almalkawi, Manel Guerrero Zapata, Jamal N Al-Karaki, and Julian Morillo-Pozo. Wireless multimedia sensor networks: current trends and future directions. *Sensors*, 10(7):6662–6717, 2010.

[72] Hasan Ali Khattak, Zoobia Ameer, Ikram Ud Din, and Muhammad Khurram Khan. Cross-layer design and optimization techniques in wireless multimedia sensor networks for smart cities. *Comput. Sci. Inf. Syst.*, 16(1):1–17, 2019.

[73] Changsu Suh, Young-Bae Ko, and Dong-Min Son. An energy efficient cross-layer mac protocol for wireless sensor networks. In *Asia-Pacific Web Conference*, pages 410–419. Springer, 2006.

[74] Ian F Akyildiz, Mehmet C Vuran, and Ozur B Akan. A cross-layer protocol for wireless sensor networks. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 1102–1107. Ieee, 2006.

[75] Lodewijk Van Hoesel, Tim Nieberg, Jian Wu, and Paul JM Havinga. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wireless Communications*, 11(6):78–86, 2004.

[76] Sennur Ulukus, Aylin Yener, Elza Erkip, Osvaldo Simeone, Michele Zorzi, Pulkit Grover, and Kaibin Huang. Energy harvesting wireless communications: A review of recent advances. *IEEE Journal on Selected Areas in Communications*, 33(3):360–381, 2015.

[77] Hafiz Husnain Raza Sherazi, Luigi Alfredo Grieco, and Gennaro Boggia. A comprehensive review on energy harvesting mac protocols in wsns: Challenges and tradeoffs. *Ad Hoc Networks*, 71:117–134, 2018.

[78] Zhi Ang Eu, Hwee-Pink Tan, and Winston KG Seah. Design and performance analysis of mac schemes for wireless sensor networks powered by ambient energy harvesting. *Ad Hoc Networks*, 9(3):300–323, 2011.

[79] Zhi Ang Eu and Hwee-Pink Tan. Probabilistic polling for multi-hop energy harvesting wireless sensor networks. In *2012 IEEE international conference on communications (ICC)*, pages 271–275. IEEE, 2012.

[80] Chisato Fujii and Winston KG Seah. Multi-tier probabilistic polling in wireless sensor networks powered by energy harvesting. In *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 383–388. IEEE, 2011.

[81] Hmidi Zohra, Laid Kahloul, and Saber Benharzallah. Using priced timed automata for the specification and verification of csma/ca in wsns. *International Journal of Information and Communication Technology*, 17(2):129–145, 2020.

[82] Paolo Ballarini and Alice Miller. Model checking medium access control for sensor networks. In *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (isola 2006)*, pages 255–262. IEEE, 2006.

[83] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In *International Conference on Computer Aided Verification*, pages 202–215. Springer, 2004.

[84] Alexandre David, Kim G Larsen, Axel Legay, Marius Mikučionis, and Danny Bøgsted Poulsen. Uppaal smc tutorial. *International journal on software tools for technology transfer*, 17(4):397–415, 2015.

[85] Alexandre David, Kim G Larsen, Axel Legay, Marius Mikučionis, Danny Bøgsted Poulsen, Jonas van Vliet, and Zheng Wang. Statistical model checking for networks of priced timed automata. In *International conference on formal modeling and analysis of timed systems*, pages 80–96. Springer, 2011.

[86] Thomas Hérault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In *International Workshop on Verification, Model Checking, and Abstract Interpretation*, pages 73–84. Springer, 2004.

[87] Hakan Lorens Samir Younes. *Verification and planning for stochastic processes with asynchronous events*. Carnegie Mellon University, 2004.

[88] Amitabha Roy and K Gopinath. Improved probabilistic models for 802.11 protocol verification. In *International Conference on Computer Aided Verification*, pages 239–252. Springer, 2005.

[89] Winston KG Seah, Zhi Ang Eu, and Hwee-Pink Tan. Wireless sensor networks powered by ambient energy harvesting (wsn-heap)-survey and challenges. In *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, pages 1–5. Ieee, 2009.

[90] Sujesha Sudevalayam and Purushottam Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys & Tutorials*, 13(3):443–461, 2010.

[91] Habib F Rashvand, Ali Abedi, Jose M Alcaraz-Calero, Paul D Mitchell, and Subhas Chandra Mukhopadhyay. Wireless sensor systems for space and extreme environments: A review. *IEEE Sensors Journal*, 14(11):3955–3970, 2014.

[92] Pardeep Kaur, BS Sohi, and Preeti Singh. Recent advances in mac protocols for the energy harvesting based wsn: A comprehensive review. *Wireless Personal Communications*, 104(1):423–440, 2019.

[93] Fabio Iannello, Osvaldo Simeone, and Umberto Spagnolini. Medium access control protocols for wireless sensor networks with energy harvesting. *IEEE Transactions on Communications*, 60(5):1381–1389, 2012.

[94] Xenofon Fafoutis and Nicola Dragoni. Odmac: An on-demand mac protocol for energy harvesting-wireless sensor networks. In *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 49–56, 2011.

[95] Xenofon Fafoutis and Nicola Dragoni. Analytical comparison of mac schemes for energy harvesting—wireless sensor networks. In *2012 Ninth International Conference on Networked Sensing (INSS)*, pages 1–6. IEEE, 2012.

[96] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320, 2006.

[97] Xenofon Fafoutis, Alessio Di Mauro, and Nicola Dragoni. Sustainable medium access control: implementation and evaluation of odmac. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 407–412. IEEE, 2013.

[98] Xenofon Fafoutis, Alessio Di Mauro, Charalampos Orfanidis, and Nicola Dragoni. Energy-efficient medium access control for energy harvesting communications. *IEEE transactions on consumer electronics*, 61(4):402–410, 2015.

[99] Neha Trivedi, G Kuamr, and Teena Raikwar. Survey on mac protocol for wireless sensor network. *International Journal of Emerging Technology and Advanced, Engineering*, 3:558–562, 2013.

[100] Ingook Jang, Dohoo Pyeon, Sunwoo Kim, and Hyunsoo Yoon. A survey on communication protocols for wireless sensor networks. *Journal of Computing Science and Engineering*, 7(4):231–241, 2013.

[101] Ahmed Akl, Thierry Gayraud, and Pascal Berthou. A metric for evaluating density level of wireless sensor networks. In *2011 IFIP Wireless Days (WD)*, pages 1–3. IEEE, 2011.

[102] Peng Lin, Chunming Qiao, and Xin Wang. Medium access control with a dynamic duty cycle for sensor networks. In *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No. 04TH8733)*, volume 3, pages 1534–1539. IEEE, 2004.

[103] Yanjun Sun, Omer Gurewitz, and David B Johnson. Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 1–14, 2008.

[104] Xenofon Fafoutis. *Medium Access Control in Energy Harvesting: Wireless Sensor Networks*. Technical University of Denmark, Applied Mathematics and Computer Science, 2014.

[105] Chulsung Park, Kanishka Lahiri, and Anand Raghunathan. Battery discharge characteristics of wireless sensor nodes: An experimental analysis. In *2005 Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005.*, pages 430–440. Citeseer, 2005.

[106] Zohra Hmidi, Laid Kahloul, Saber Benharzallah, and Nadia Hamani. Performance evaluation of odmac protocol for wsns powered by ambient energy. *International Journal of Simulation and Process Modelling*, 17(1):67–78, 2021.

[107] Zohra Hmidi, Laid Kahloul, and Saber Benharzallah. A new mobility and energy harvesting aware medium access control (meh-mac) protocol: Modelling and performance evaluation. *Ad Hoc Networks*, 142:103–108, 2023.

[108] Ricardo Silva, Jorge Sá Silva, and Fernando Boavida. Mobility in wireless sensor networks–survey and proposal. *Computer Communications*, 52:1–20, 2014.

[109] Syed Wajahat Abbas Kazmi, Adrian Kacso, and Roland Wismüller. Recent mac protocols for mobility-aware wireless sensor networks—a survey and future directions. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 159–164. IEEE, 2017.

[110] Mahdi Zareei, AKM Muzahidul Islam, Cesar Vargas-Rosales, Nafees Mansoor, Shidrokh Goudarzi, and Mubashir Husain Rehmani. Mobility-aware medium access control protocols for wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 104:21–37, 2018.

[111] Romain Kuntz and Thomas Noël. Machiavel: Accessing the medium in mobile and dense wsn. In *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1088–1092. IEEE, 2009.

[112] Romain Kuntz, Julien Montavont, and Thomas Noël. Improving the medium access in highly mobile wireless sensor networks. *Telecommunication Systems*, 52(4):2437–2458, 2013.

[113] Georgios Z Papadopoulos, Antoine Gallais, Thomas Noel, Vasileios Kotsiou, and Periklis Chatzimisios. Enhancing contikimac for bursty traffic in mobile sensor networks. In *SENSORS, 2014 IEEE*, pages 257–260. IEEE, 2014.

[114] A Dunkels. The contikimac radio duty cycling protocol, sics technical report t2011: 13, 2011.

[115] Papa Dame Ba, Bamba Gueye, Ibrahima Niang, and Thomas Noel. Mox-mac: A low power and efficient access delay for mobile wireless sensor networks. In *2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)*, pages 1–6. IEEE, 2011.

[116] Vivek Kumar Verma and Vinod Kumar. Review of mac protocols for energy harvesting wireless sensor network (eh-wsn). In *Internet of Things and Big Data Applications*, pages 141–149. Springer, 2020.

[117] Seong Cheol Kim, Jun Heon Jeon, and Hyun Joo Park. Qos aware energy-efficient (qaee) mac protocol for energy harvesting wireless sensor networks. In *International Conference on Hybrid Information Technology*, pages 41–48. Springer, 2012.

[118] Hsiang-Ho Lin, Mei-Ju Shih, Hung-Yu Wei, and Rath Vannithamby. Deepsleep: Ieee 802.11 enhancement for energy-harvesting machine-to-machine communications. *Wireless Networks*, 21(2):357–370, 2015.

[119] Thien D Nguyen, Jamil Y Khan, and Duy T Ngo. An adaptive mac protocol for rf energy harvesting wireless sensor networks. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[120] Adnan Ismail Al-Sulaifanie, Subir Biswas, and Bayez Khorsheed Al-Sulaifanie. Ah-mac: adaptive hierarchical mac protocol for low-rate wireless sensor network applications. *Journal of Sensors*, 2017, 2017.

[121] Kien Nguyen, Vu-Hoang Nguyen, Duy-Dinh Le, Yusheng Ji, Duc Anh Duong, and Shigeki Yamada. Eri-mac: An energy-harvested receiver-initiated mac protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 10(5):514169, 2014.

[122] Demin Gao, Shuo Zhang, and Fuquan Zhang. Has-mac: A hybrid asynchronous and synchronous communication system for energy-harvesting wireless sensor networks. *Wireless Personal Communications*, 119(2):1743–1761, 2021.

[123] E-YA Lin, Jan M Rabaey, and Adam Wolisz. Power-efficient rendez-vous schemes for dense wireless sensor networks. In *2004 IEEE international conference on communications (IEEE Cat. No. 04CH37577)*, volume 7, pages 3769–3776. IEEE, 2004.

[124] Sara Khalifa, Guohao Lan, Mahbub Hassan, Aruna Seneviratne, and Sajal K Das. Harke: Human activity recognition from kinetic energy harvesting data in wearable devices. *IEEE Transactions on Mobile Computing*, 17(6):1353–1368, 2017.

[125] Ledeng Huang, Ruishi Wang, Zhenhua Yang, and Longhan Xie. Energy harvesting backpacks for human load carriage: Modelling and performance evaluation. *Electronics*, 9(7):1061, 2020.

[126] FR Pathan. Kinetic energy harvesting from human hand movement by mounting micro electromagnetic generator. In *E3S Web of Conferences*, volume 115, page 02005. EDP Sciences, 2019.

[127] Michele Magno, Dario Kneubühler, Philipp Mayer, and Luca Benini. Micro kinetic energy harvesting for autonomous wearable devices. In *2018 International symposium on power electronics, electrical drives, automation and motion (SPEEDAM)*, pages 105–110. IEEE, 2018.

[128] Marco Zuniga and Bhaskar Krishnamachari. Analyzing the transitional region in low power wireless links. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, pages 517–526. IEEE, 2004.

[129] Ruihong Wang, Wuyungerile Li, Fei Gao, and Taofeng Jiao. The psl mac protocol for accumulated data processing in the energy-harvesting wireless sensor network. *Wireless Communications and Mobile Computing*, 2022, 2022.

[130] PK Dash and M Panda. Smlmac-heap: Slotted multi-layer mac protocol for wireless sensor networks powered by ambient energy harvesting. *Indian Journal of Science and Technology*, 15(36):1827–1835, 2022.

[131] M Rajesh, BL Raju, and BN Bhandari. Mobility aware adaptive hybrid mac protocol for wsn. *International Journal of Wireless Information Networks*, 29(2):157–166, 2022.

[132] VC Diniesh and G Murugesan. Eem-mac: Enhanced energy efficient mobility aware mac protocol for mobile internet of things. *Peer-to-Peer Networking and Applications*, pages 1–20, 2022.

[133] Arif Obaid, Xavier Fernando, and Muhammad Jaseemuddin. A mobility-aware cluster-based mac protocol for radio-frequency energy harvesting cognitive wireless sensor networks. *IET Wireless Sensor Systems*, 11(5):206–218, 2021.