UNIVERSITÉ DE STRASBOURG

École doctorale MSII

Contributions to Low Power Listening in Dynamic Internet of Things

MÉMOIRE

présenté

pour l'obtention d'une

Habilitation à diriger des recherches
de l'Université de Strasbourg
(mention informatique)

par

Antoine Gallais

Composition du jury

Président :	Marcelo Dias de Amorim (DR CNRS)	LIP6 - UPMC, Paris
Rapporteurs :	André-Luc Beylot (PU)	IRIT - ENSEEIHT, Toulouse
	Silvia Giordano (Professor)	SUPSI - University of Applied Science, Suisse
	Fabrice Valois (PU)	CITI - INSA Lyon
Examinateurs :	Nathalie Mitton (DR Inria)	Inria Lille - Nord Europe
	Thomas Noël (PU, garant)	ICube - Université de Strasbourg

Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie
ICube (UMR 7357)

Mis en page avec la classe thesul.

Contents

Chapter 1

Introduction

Contents

1.1 Why this document

I am presenting this document in the context of the *habilitation à diriger des recherches* diploma. It provides details on most of the research problems I addressed since September 2008, when I joined the University of Strasbourg as an associate professor, back in with several colleagues of mine. This overview covers the solutions that have been proposed, including ours and also the perspectives and ambitions that I have currently. All have been cross-fed by the students I was glad to advise (e.g., M1, M2, engineer, PhD) and the research colleagues I was fortunate to work with (e.g., local research group, national and international collaborations, national and international research projects). They have been fostered also by the interactions I had as a teacher and incidentally as the coordinator of the second year of our Master in Computer Science (networks and embedded systems speciality, from 2009 to 2015) and of our professional bachelor degree in administration of computer networks (from 2014 to 2017). Those experiences allowed to enrich my understanding and knowledge about our field and some of its operational aspects, while confronting myself to various audiences (DUT, Licence, Master and engineering schools). They also brought me closer to the reality of the industry as I could better apprehend the constraints and possibilities for innovations, especially around the Internet of Things and more generally in computer networks. A detailed resume is provided in Appendix **??** (in french).

1.2 Research context

1.2.1 From ad hoc networks to (industrial) Internet of Things

Wireless devices have played an increasingly important role in our digital lives. Over the last decade, smartphones, tablet PCs, as well as many other uniquely identifiable devices have gained much attention especially as enhanced Internet access service was allowing them to be connected from nearly everywhere.

In the early 2000s, mobile *ad hoc* networks (MANET) started to be standardized (e.g., `manet` working group at IETF . Multihop wireless networks were being investigated intensely, with numerous contributions at every layer of the communication stack (e.g., antennas, radio channel, IPv6 over low-power wireless personal area networks, routing over lossy links). While operating wireless networks without any fixed or a priori deployed infrastructure was becoming feasible, the constantly growing number of connected devices had started to shape the Internet of Things (IoT) paradigm.

Among those devices, wireless sensors were emerging. These highly constrained wireless nodes (e.g., limited transmission range, processing and storage capability, battery-powered) were able to sense their environment while processing and transmitting the collected data (e.g., light, temperature, humidity, motion, sound, pressure) in a wireless manner. Deployments of such devices over an area of interest were giving birth to Wireless Sensor Networks (WSN).

Since then, the number of *things* (e.g., machines, appliances, computers, humans) able to store, process and transmit data has exceeded the number of people on Earth[1]. The intensive research and development activity on the interconnection of such things has had the Internet of Things (IoT) as a logical consequence. It has led to an unprecedented set of control and monitoring applications [26–28]. Such IoT deployments and multiple instances of information and communication technology (ICT) have even been gathered in a secure fashion to manage the assets of cities (e.g., schools, transportation systems, hospitals, power plants, water supply networks, waste management), thus enabling the concept of smart city.

Radio transmissions have even tended to replace reliable wired connections, as modern applications are more and more demanding in terms of responsive communications with high reliability. We have now a large variety of radio technologies (e.g., IEEE802.15.4, 6TiSCH, WirelessHART, WiFi, Bluetooth, LoRa, Z-Wave) that allows to connect more and more smart objects. The so-called Industrial Internet of Things (IIoT) aims at transforming all the devices for the supply and manufacturing chain into autonomous radio embedded devices. This has led to a large variety of applications whose successful deployments have outshone the difficulties of sharing a common resource (air) among many heterogeneous objects, in a distributed manner. Consequently, research studies have continuously aimed at optimizing the network performance in terms of lifetime (i.e., energy saving schemes), reliable data collection, congestion avoidance, or quality of service for the user. Among all remaining research problems, avoiding the unnecessary energy wastage at each sensor keeps requiring energy-efficient communication protocols.

[1]Source: Cisco (2008), `http://blogs.cisco.com/news/the-internet-of-things-infographic/`

1.2.2 Why controlling medium access ?

Reduced transmission ranges (for energy saving purposes) impose multihop transmissions within the network that lead to interferences and time-varying radio link quality.

WSNs have been considered as Low-power and Lossy Networks (LLNs), which consist in a large number of spatially distributed autonomous sensor nodes. They are assumed to disseminate their readings by cooperatively sending, receiving and forwarding measurements from other nodes. Those data are intended to one or more Internet connected gateways (i.e., sinks) which may also process or store the received readings.

Several types of communication paradigms are considered in a WSN, namely time-driven, event-driven and query-driven. Time-driven applications induce regular transmissions from deployed nodes to the sink(s). Event-driven refers to sensor devices sending data upon event detection only while query-driven networks trigger data collect from end user's demand. Deployments with hybrid requirements have also been observed (e.g., [27, 29]).

In most of applications, data traffic is considered either as multipoint-to-point (i.e., convergecast), point-to-multipoint (i.e., a node sending requests to one or more devices) or point-to-point (i.e., sensor-to-sink communication, peer-to-peer transmissions within the network).

Regardless of the traffic nodes have to support, wireless devices must share air as a communication medium. Any transmitting node can use the resource safely provided that no other surrounding nodes is using it simultaneously. In such networks, among all operations of a sensor node, the wireless communication is the one inducing the most energy consumed [30], mainly due to idle listening (nodes sampling the medium without any incoming transmission occurring), overhearing (nodes listening to transmissions intended for other surrounding devices), and retransmissions (collisions upon simultaneous data transmissions in a vicinity).

Consequently, we decided to investigate the Medium Access Control (MAC) layer whose duty is to allow each node to coordinate its own actions related to the wireless medium. Indeed, MAC is responsible for alternating the radio between periods of activity (transmitting and receiving a packet) and sleep mode. Two duty-cycled nodes can communicate with each other if and only if both are active at the same time. Medium access must be controlled in order to reduce energy consumption while preserving network performances (e.g., latency, throughput).

Such control in wireless operations had already been much investigated in early wireless standards (e.g., IEEE 802.11 standard [31]). Those initiatives however assumed that wireless nodes could present some (high) computation resources, memory and synchronization hardware, as well as unlimited energy resources. Consequently, they are not considered as potential candidates for dynamic multihop networks whose energy efficiency is one of the main requirements.

1.2.3 How to control medium access ?

Numerous MAC layer solutions have thus been dedicated to WSN. As brought up in [32], many problems were remaining open from the MAC layer perspective. While the so-called slotted schemes had to determine communication schedules for each node (e.g., either centrally from an omniscient sink or locally at each device in a distributed manner), asynchronous protocols have allowed for more flexible approaches where time synchronization among nodes is not mandatory.

Time Division Multiple Access (TDMA) based schemes. The allocation of time slots to wireless nodes aims at guaranteeing a collision-free schedule, thus being particularly suitable to handle periodic traffic. Outside its time slots, a node can turn its radio off. While a vast range of contributions followed the seminal works (e.g., Self-Organizing MAC for Sensor

Networks [33], FLow-Aware Medium Access [34], Multi-Frequency MAC for WSNs [35], Traffic-adaptive MAC [36]), some dedicated standards have also emerged (e.g., IEEE802.15.4-2006, IEEE802.15.4-2015). They aim at combining TDMA with slow channel hopping in order to enable reliability and energy efficiency while combating narrow band noise, very frequent in industrial environments.

Protocols with common active periods. Nodes are assumed to be time synchronized and continuously alternate between active and sleep periods. Periodical wake-ups allow to transmit or receive data by using contention-based schemes (e.g., CSMA). Such protocols appear more appropriate for applications dealing with time-driven traffic (e.g., monitoring). The initial contribution (i.e., S-MAC [37] was followed by several others (e.g., Timeout MAC [38], Separate Wakeup MAC [39]). The IEEE 802.15.4 standard [40] allows nodes to compete for the medium in a contention-access period (i.e., slotted-CSMA) within each time slot. Such active periods are signaled by beacon messages while energy is saved by nodes switching their radio off before the next beacon transmission (inactive period).

Preamble-sampling protocols do not require any synchronization and rely on low-power listening (LPL) mechanisms. Indeed, nodes asynchronously sample the wireless medium for incoming packets at regular intervals. In between, their radio is turned off to reduce energy consumption. The scheduling of a node is therefore determined independently of other nodes. Any transmitter must send a preamble, prior the data packet, to ensure that the intended receiver will stay on upon sampling the medium. The preamble duration must be at least as long as the sampling period of the receiving node. Preamble-Sampling ALOHA [41], Preamble-Sampling CSMA [42] and B-MAC [43] were among the first of the preamble-sampling protocols.

Hybrid protocols combine the concepts of synchronized and asynchronous protocols. Such solutions allow to switch from a scheduled TDMA-based scheme (e.g., common active/sleep S-MAC) to CSMA schemes, based on observed contention levels. Zebra MAC (Z-MAC) [44], Scheduled Channel Polling (SCP) [45], MH-MAC [46] and Funneling-MAC [47] belong to this family of protocols. For instance, in [47], an hybrid CSMA/TDMA algorithm is used on nodes around the sink while a pure CSMA scheme (e.g., B-MAC) is deployed on remaining nodes.

When contemplating envisioned applications, we anticipated that dynamics would play a major role in IoT networks (e.g., varying traffic, mobile nodes). We thus aimed at flexible protocols with low complexity (i.e., avoid costly time synchronization and scheduling) and decided to focus on asynchronous and LPL-based solutions. No scheduling (and distribution) was needed for the communication between two sensor nodes [48] and a large number of contention-based solutions had emerged over a short timespan [49]. Moreover, as surveyed in [50], several IoT networks were relying on preamble-sampling MAC protocols while multiple issues were yet to be addressed. Especially, we focused on envisioned IoT applications and deployments whose dynamics would require heterogeneous low-power listening.

1.2.4 Why focusing on heterogeneous low-power listening ?

Back in 2009, after surveying existing deployments of WSN, we identified two major and distinct categories of applications that were matching two communication paradigms, namely time-driven (e.g., habitat monitoring [51], glacier observation [52]) and event-driven (e.g., structural health monitoring [53]). The former were imposing nodes to operate with low duty-cycle and low-power consumption, a thus inducing small sampling rate and low data rates. The latter were requiring high data rates, high fidelity sampling (through a reliable end-to-end protocol), precise time-stamping and hence efficient time synchronization.

This study had allowed us to identify some common characteristics among those applications that could have an impact on the MAC layer [1]. As summarized in Table 1.1, applications were tolerant to delays as most of the collected data was analyzed once the experiment was completed. The observed topologies were involving quite few nodes (of the order of tens of sensors) which were carefully placed and configured (radio range) to obtain the required connectivity from startup. Most of them were single-hop (sometimes a few hops, but hardly up to 6 hops).

Characteristics	**as of 2009**	**Impact on the MAC layer**
Tolerance to delay	High tolerance	Best effort schemes are adequate
Size of the network	Relatively small	Small collision domain
Topology	Single to few hops	Simplified link management
Sensor placement	Carefully studied	Simplified neighborhood management
Length of deployment	Less than a year	Energy management is a secondary concern
Mobility	Few number of nodes	Simplified scheduling scheme

Table 1.1: Main characteristics of WSN deployments as observed in 2009.

In addition, few deployments of mobile WSN had been proposed while those depicted in [29, 54] were presenting only five and seven mobile sensors respectively. Mobility was foreseen as a very likely solution to expand network coverage, improve the routing performance or the overall connectivity [55]. In every mobile scenario, the small number of nodes or the low frequency of communications only increased slightly the complexity of the deployments.

Even though several deployments had been successful, design choices were very application-dependent and most of the communication stacks were being built from scratch, thus making the reuse of these solutions difficult. In turn, these potential difficulties led to more and more MAC layers built from scratch. To our minds, this vicious circle was raising two major drawbacks. First, it was imposing WSN deployments to be performed by networking experts with strong programming skills in embedded systems. Second, the implemented solutions would become barely usable once confronted to slightly different deployment constraints. Furthermore, the limited durations of deployments or the use of energy harvesting mechanisms (e.g., solar panels) had certainly kept engineers from optimizing the communication stack in term of energy consumption. For example, even though Sensorscope [52] had been a successful long-term deployment, its 10% duty-cycle would have been considered as excessive in WSN [56].

Several low-power listening based MAC protocols had been presented well adapted to pre-configure the MAC layer of the sensor nodes prior to ease the configuration of some radio-related parameters (e.g., sampling period, preamble length) and thus turn into versatile solutions. Two contributions quickly emerged from the literature, namely B-MAC and X-MAC (Figure 1.1).

While B-MAC was requiring a full preamble before any data transmission, protocols such as X-MAC [57], MX-MAC [58], SpeckMAC [59] or ContikiMAC [60] had replaced this costly preamble with series either of strobes (e.g., X-MAC) or of data packets (e.g., ContikiMAC). Hence, the transmitter repeatedly was sending strobes or data that contained the address of the receiver. On the other side, the intended receiver could reply with an acknowledgment while staying on until the data transmission was complete. Nodes that had a different address from the one that is indicated in the strobe could go back to sleep. Moreover, with BoX-MACs [61] or ContikiMAC, a node could be configured so that to keep the radio on for short time, right

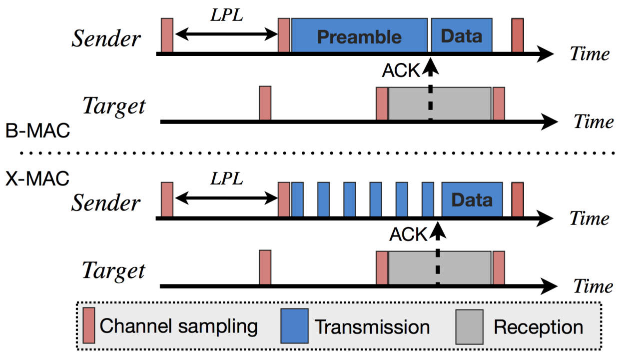


Figure 1.1: B-MAC and X-MAC transmission patterns.

after receiving a packet in order to cope with consecutive packets. These protocols were reducing overhearing and achieved low power operation at both the receiver and transmitter sides.

Most importantly, all those protocols introduced interfaces in order to tune the LPL values. Such configuration could be made either offline (i.e., prior tot he deployment) or at runtime (i.e., during the deployment). These schemes could thus match various requirements with a single proposal. For instance, time-driven and event-driven networks would have performed very differently on a single and monolithic MAC layer. The location of a given event being unpredictable, any prior installation of a specific MAC layer on nodes close to the spot of interest had been excluded.

However, these available solutions did not meet our goals of automatic a priori and on-the-fly localized adaptation of duty-cycle configurations in the network, since all the nodes of the network were operating homogeneously. In addition, their operation in a dynamic scenario would have required periodic re-computation of the adjustable parameters while the network was running.

Eventually, while aiming at fully automated setup of the medium access protocol, we were also considering heterogeneous LPL configurations cohabit in the network, knowing that any unfortunate modification could isolate a node, or worse, create a partition in the network. Finely negotiated configurations would rather lead to use a unique dedicated protocol for applications with varying demands (e.g., traffic) over unstable topologies (e.g., mobile nodes).

1.3 Challenges addressed in this document

Since 2008, most of my research activities have been focused on MAC layer in IoT networks, and more specifically on preamble-sampling approaches. Despite the quick emergence of synchronized (and standardized) methods, we felt many unanswered questions around LPL-based were in tune with most of our target applications (e.g., remote healthcare, wildlife monitoring). We thus focused on three challenges, namely a priori configuration of heterogeneous LPL values, runtime adaptation of preamble and sampling periods and communications of mobile nodes with a static infrastructure of low-power listening devices.

A priori automatic configuration of low-power listening. Most of existing solutions were relying on a priori configured LPL devices, as envisioned applications were inducing anticipated traffic patterns and relying on static topologies. We aimed at introducing heterogeneity at the MAC layer, without endangering network connectivity and, most importantly, while considering application and network requirements. We proposed several methods that allowed for automatic a priori configuration of LPL values, based on physical and routing topology (i.e., LMST construction, tree-based decisions) or application requirements (i.e., activity scheduling, sleep depth). Our results were obtained during the supervision of Dr. Julien Beaudaux.

Runtime adaptation of low-power listening. After investigating a priori configuration of LPL values, we studied the runtime adaptation of these parameters. We had indeed observed that IoT deployments were evolving very quickly, while combining different application paradigms (i.e., event-driven and time-driven) and thus inducing non predictable traffic patterns (i.e., burst transmissions). The identified challenge was to enable energy-efficient links (i.e., small sampling and preamble periods) from specific locations, upon event occurrence, while maintaining a mostly sleeping network. We proposed several mechanisms that enabled implicit negotiation of energy-efficient links while mitigating packet duplications. These works were mostly initiated during the PhD period of Dr. Romain Kuntz and were pursued with the supervision of Dr. Georgios Z. Papadopoulos.

Handling mobility in IoT. As the Internet of Things was encompassing a large set of applications and deployment scenarios, we observed that medium access control of devices moving around a static infrastructure was still an open issue. We proposed to enhance the well-known ContikiMAC protocol in order to enable mobile-to-static communications. We also introduced some mechanisms for efficient neighbor discovery and improved handovers. Our results were obtained during the supervision of Dr. Georgios Z. Papadopoulos, and led to collaborations with Vassilios Kotsiou and Dr. Periklis Chatzimisios.

The proposed mechanisms and protocols were being tested under both both simulations and experimentation environments. To finely anticipate the impact of a versatile networking protocol, a key feature of the available testbeds was their ability to emulate real environments. In this context, all our experimentations have been conducted over the FIT IoT-lab project[2], which has been built throughout the considered 2008-2017 period.

1.4 Structure of this document

This document is organised as follows. Chapter 2 summarizes our works towards a priori configuration of heterogeneous LPL values based on several inputs (e.g., application, routing) and methods (e.g., LMST, sleep depth). Chapter 3 covers our contributions to LPL mechanisms facing dynamics, and more specifically non uniform traffic over time (e.g., burst transmissions). Chapter 4 deals with mobile nodes having to gain access to the medium in order to offload their data to the sink station, by using an infrastructure of static nodes. Eventually, Chapter 5 provides concluding remarks and envisioned research perspectives.

[2](initially known as SensLab project) https://www.iot-lab.info/

Chapter 2

Self-configuration of low-power listening

Contents

2.1 Research context

As previously discussed in Section 1.2.4, LPL parameters (preamble length and sampling period) used to be strongly related to some application requirements and ensued traffic pattern. Nodes configured with short sampling periods would be able to manage more traffic load while requiring short preambles for their data transmission. However, due to more frequent radio wake-ups, such benefits would be at the cost of increased energy consumption. Conversely, longer preamble and sampling periods would reduce energy consumption while preventing nodes from handling high traffic load. However, most of existing solutions were affecting LPL configuration once and for all, prior to the deployment. Moreover, it was often considered as homogeneous over the set of nodes. Such an approach failed to satisfy the requirements of dynamic applications (e.g., varying data traffic) or even the ones of simpler event-driven scenarios (e.g., data traffic originating from unpredictable areas).

Few studies had thus introduced heterogeneity of LPL configurations for power management in IoT networks. Yet, In order to fit with evolving deployment conditions, heterogeneous LPL configuration would have been much appreciated, even though the coexistence of several configurations was raising new issues (e.g., isolated nodes, reduction of network performances).

Kansal et al. [62] presented a prediction of energy evolution (consumed and gained energy with solar-powered sensor nodes) to determine the optimal duty-cycling of each node, from the application standpoint. However, no MAC layer duty-cycling was considered.

In [63] dynamic heterogeneous radio duty-cycling was addressed in energy harvesting networks. The proposed mechanisms considered residual energy (i.e., remaining energy in the battery) or prospective increase of energy (i.e., expected energy gain from harvesting) in order to define the duty-cycle configuration of each node. However, the sole use of energy as a criterion for scheduling of the radio activity could worsen networks performances (e.g., no consideration of node location, neither of ongoing traffic).

Several methods relied on learning phases during the early stages of the network deployment. WiseMAC [64] allowed nodes to overhear ongoing transmissions in order to learn the sampling periods of their neighbors, thus avoiding any prior costly preamble. The duration of the learning phase was however unbounded as it was depending on data traffic. While any change of topology would require to reset the learning process in order to discover the new LPL values within the vicinity, any traffic burst would have led to congestion as LPL parameters of neighboring nodes could not be negotiated.

DutyCon [65] and pTunes [66] used control theory to determine LPL values at each node, by using ongoing data streams and performance constraints a priori defined by the end user (e.g., minimal delay, maximal loss rate, expected lifetime). Each node could include link parameters (e.g., traffic load, delays) in its messages towards the sink which would confront those metrics to the pre-defined constraints in order to determine the most appropriate LPL values. Newly defined parameters were further broadcasted to the whole network. Those propositions assumed centralized processes and required central entities that maintain a global view of the network. Even though piggybacking control information within data packets helped limiting the overhead, broadcasting LPL values throughout the network imposed an additional synchronized mechanism.

2.2 Self-determination of LPL configurations

While heterogeneous LPL configurations may endanger network operations (e.g., ensued isolated nodes), a careful assignment of LPL values was much needed. We here detail how we managed to enhance low-power listening solutions in order to allow for automatic configuration of the LPL parameters during the initial steps of a deployment.

During the PhD period of Julien Beaudaux, we aimed at deriving LPL values from the status of each node. This status had to depend on the role played by the device for the whole network (e.g., required for routing, required for sensing). Indeed, as observed from the literature, sampling periods should have been linked to anticipated traffic. Thus, relay or sensing-only nodes (i.e., transmitting their own data only) should self-determine different LPL values. Our objective was to enable such decisions in a fully decentralized manner, thus limiting the ensued control overhead. Consequently, we investigated different mechanisms that would allow to divide the set of nodes into disjointed subsets, each corresponding to given LPL configurations.

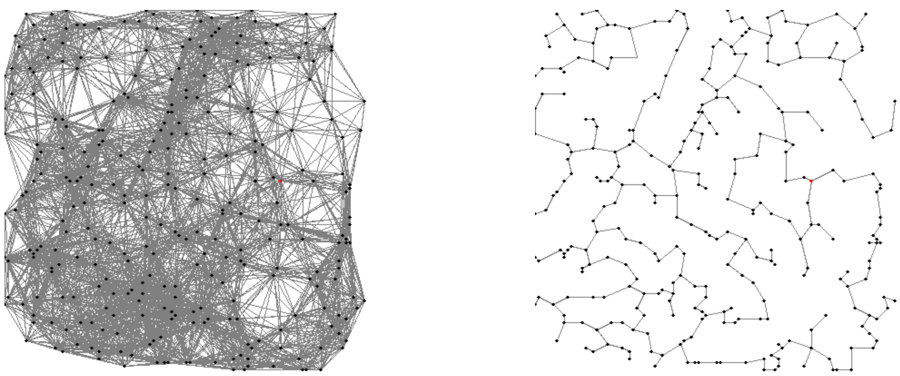
Each node had to self identify its activity state, and thus the subset it would belong to, according to a criterion dependent of the deployment requirements. Among all eligible criteria for the profiling step, we here detail two that appeared particularly suited, namely the routing role and the sensing requirement of each node.

During my PhD period, I had investigated activity scheduling and minimal sets of active nodes to achieve connected area coverage. Right after these contributions, we observed that, among these active nodes, some could spare themselves the trouble of participating in the routing process without endangering the information relaying towards the sink [2]. Indeed, maintaining global connectivity through local decisions at each node was made through a quite restrictive local criterion (i.e., for each node, the set of neighbors should be connected), thus leaving much room for further reductions of nodes participating in the multihop communicating process. Consequently, we proposed to introduce a new activity state, called sensing-only, that should be assigned to nodes that do not participate in the routing process (i.e., only transmit their own information to the sink).

Finding the maximal subset of nodes unnecessary to guarantee the connectivity of the network can be summarized as finding its Maximum Leaf Spanning Tree (MLST), which is known to be a NP-complete problem. Consequently, we proposed several localized approximations, ranging from the use of the physical topology (i.e., graph-based techniques) to the derivation of application requirements while also considering information from the routing operations [3]. Finally, we introduced the concept of sleep depth, that allows multiple LPL configurations to coexist in a single network, with minimal control overhead and guaranteed connectivity and coverage for routing and application purposes respectively.

2.2.1 Deriving LPL configurations from the physical topology

We first aimed at distinguishing nodes based on a self-determined role. We opted for local minimum spanning trees and compared ourselves with several other techniques (e.g., Voronoï diagrams, distributed greedy algorithms). As locally finding the MST of a topology is known to be a NP-complete problem, we proposed to approximate it by rely on the local minimum spanning tree (LMST) algorithm, proposed in [67].



(a) Initial topology (b) Resulting topology

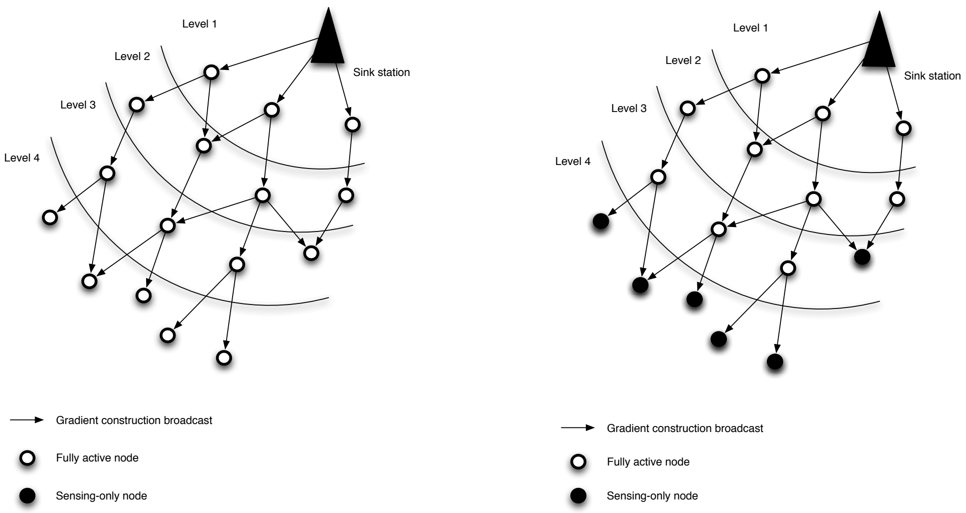
Figure 2.1: Tree approximation using a LMST

Each node had to compute the MST of the topology (i.e., its neighbors and itself), while an edge was added to the global topology if its two ends had agreed on its existence (i.e., edge appearing in their respective computed MST). We could identify sensing-only nodes as those with only one outgoing edge. These leaves could consequently stop participating in multihop communications. An example of initial WSN and resulting MST approximation topologies are provided in Figure 2.1. The construction of the LMST itself theoretically requires O(n) messages, and an average of three messages per nodes.

2.2.2 Using the role played in the routing topology

Before a network can start operating, both the physical and logical topologies must be established. While the logical link layer allows point-to-point communications, the routing layer uses those connections to enable multihop transmissions. Each node is thus given a role, depending on various characteristics (e.g., capabilities, position) that determine whether it shall send its own messages only or if it should also act as a transmission relay. Consequently, we proposed to identify the subset of sensing-only by using some existing information on each node, that reflect their actual role in the network. Many routing protocols in WSN are based on the construction of Directed Acyclic Graph (DAG) [68–70]. Every node having no *son* in the routing tree can self-identify as sensing-only, without any additional control message being required.

Figure 2.2 depicts a gradient construction (i.e., every node emits a message stating at which rank it stands from the sink station) that allows each node to know the minimal distance to the sink station (i.e., number of hops here) for itself and its direct neighbors (Figure 2.2(a)). We used this information to determine a set of leaf nodes, which can set themselves in sensing-only state, without disrupting the network connectivity (Figure 2.2(b)).



(a) Gradient construction. (b) Leaves become sensing-only.

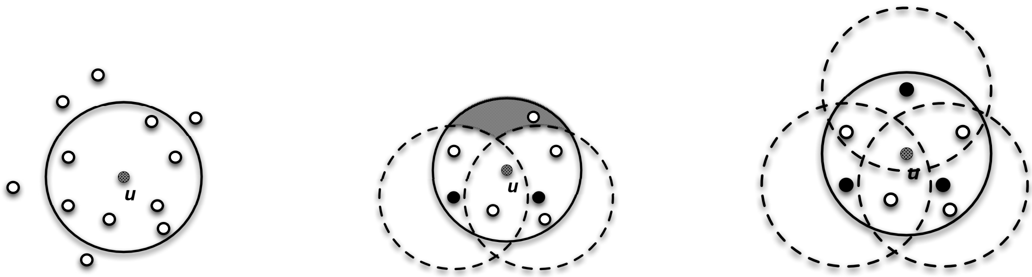
Figure 2.2: Using gradient construction to determine sensing-only nodes

2.2.3 Relying on application-based activity scheduling

Above the routing layer, stands the application layer from which multiple inputs can be gathered regarding the nodes expected activity (e.g., required monitoring density, response delays, fault-tolerance). We proposed to have LPL values depending on those requirements, and opted for intermediate activity states as defined by activity scheduling techniques [4].

Activity scheduling mechanisms aim at fulfilling the requirements of the target application with only a subset of nodes called active. The remaining nodes (passive), are then put in an energy-saving mode. Depending on the application, several criteria can be set. We considered the global sensing coverage of the area of interest, which had to be preserved by the subset of active nodes. Global connectivity of the network also had to be ensured, so that each node could send its readings to the sink, at any moment. In order to improve network lifetime (e.g., the period during which no node fails from battery exhaustion) the sensor nodes can alternate their activity mode simply by launching several activity decision rounds at regular intervals.

We proposed to associate each activity mode with some pre-defined LPL values, thus allowing for heterogeneous configuration of the whole set of nodes, without endangering network connectivity nor operation (from the application standpoint, e.g., area coverage). We based our contribution on two of the four algorithms proposed during my PhD thesis, especially because of their very low control overhead [5]. These mechanisms, respectively referred to as positive-only (PO) and positive-retreat (PR) work as follows. Every node u in the network selects a timeout. Upon timeout expiration, u evaluates its own coverage and its neighborhood connectivity, based on received decisions made by neighbors with shorter timeouts. This coverage test is positive if and only if u is fully covered and its direct neighbors are connected. This step ensures connected area coverage. If u gets passive, no message is sent. If active, u sends an activity announcement. This solution is referred to as positive-only (PO). Nodes with higher timeout values that receive the message from u will consider u during their coverage evaluation. If any sensor u has decided to be active while additional coverage is provided by other nodes with longer timeouts, u can change its decision. Once a new timeout expires, if u is covered then it sends a retreat message. This solution is referred to as positive-retreat (PR). The general principles of PO and PR are exposed in Figure 2.3.



(a) After expiration of its timeout, u has not listed any neighbor.

(b) Listed neighbors of u (black nodes) do not cover its sensing area.

(c) Listed neighbors of u (black) cover its sensing area.

Figure 2.3: Activity scheduling over the network.

Upon expiration of its timeout, a node u announces its activity if no neighbors have signaled themselves earlier (Figure 2.3(a)) or discovered neighbors do not cover the sensing area of u (Figure 2.3(b)). If known neighbors do cover its sensing area (Figure 2.3(c)), u becomes passive. If no earlier activity decision was announced, no message is sent, else u advertises its neighborhood of its new passive status, indicating its retreat. Each signaling packet contains the coordinates of the emitting node and indicates its type (i.e., positive or retreat).

Each variant can be used depending on the deployments characteristics. Indeed, while PO requires less overhead than PR, it also induces larger sets of active nodes. Yet, with PR, the loss of a retreat message may disrupt the protocol general functioning, some active nodes being potentially unaware of some other node retreat.

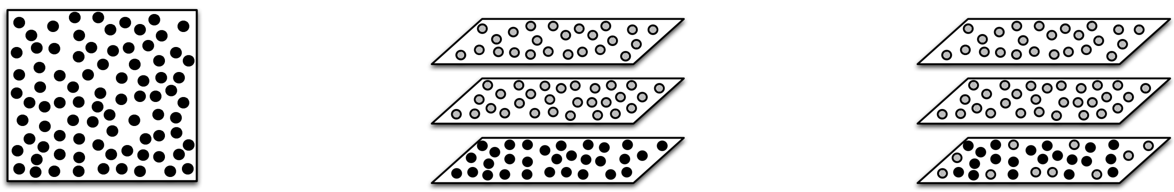
2.2.4 Introducing sleep depth

In order to allow for multiple choices when configuring LPL values, we further introduced the concept of sleep depth [6, 7]. Our objective was to partition the network into multiple disjoint sets that would each correspond to a LPL configuration, i.e., a sleep depth. We thus decided to combine the need of a minimal connected set for routing purposes (as detailed in 2.2.2) and the opportunity of various levels of activity as per the application requirements (as studied in 2.2.3).

We proposed to have LPL configurations corresponding to different sleep depths. From the application point of view, nodes could be organized in distinct activity sets based on various criteria (e.g., required coverage or connectivity degree, fault-tolerance requirements). Those would lead least required nodes to self-opt for the longest sampling period. Conversely, most needed ones would select the shortest sampling interval.

In order to transpose such application requirements into adapted LPL configurations, we adopted a layered virtual structuration among nodes. Among all possible criteria, the number of nodes within communication range (i.e., radio density) allowed each node to estimate the number of devices that might try to access the radio channel. The lower the density, the higher the probability of having to forward data, thus the shorter the duty-cycle.

We proposed to separate the nodes into different virtual layers that would further correspond to distinct sleep depths, and thus distinct LPL configurations. A threshold D_{max} could be either provided by the user as a parameter of the targeted deployment, or deduced from the deployment requirements. Known by each device, this value allowed each node to to self-determine its potential sleep depth, based on the observed radio density. During a timeout, each node processes packets from its neighbors. These messages include the selected layer of the sender thus allowing the receiver to maintain a list of its neighbors and their respective layers. Upon timeout expiration, the deciding receiver evaluates the number of its neighbors at each layer and opts for the first one whose radio density is below D_{max} (i.e., layer 0 if no message received). It then transmits a message to inform its neighborhood of its activity decision and chosen layer.



(a) Random physical topology.

(b) Three virtual layers and sensing-only identification (gray nodes) at upper layers.

(c) Routing information can allow for more sensing-only devices.

Figure 2.4: A given topology can be transformed into multiple disjoint sets of devices which either turn into active (black nodes) or sensing-only mode (gray nodes).

Then the node selects the LPL configuration that corresponds to the layer at which it has decided to be active. The LPL value is doubled for each layer below the selected one. We decided to dedicate the lowest layer to the routing backbone (i.e., shortest duty-cycle and thus highest capability to handle traffic). Each node belonging to upper layers will transmit its data to one of its neighbor of the lowest layer, that will forward it up to the sink.

Figure 2.4 illustrates this concept with three layers. The random physical topology (Figure 2.4(a)) is divided into three distincts subsets. Connectivity could only be imposed at first layer in order to allow each sensor to upload its readings towards the sink station, thus allowing sensors of upper layers to become sensing-only (Figure 2.4(b)). Finally, as detailed in Section 2.2.2, the routing operation implemented at first layer also allowed to identify leaves in the gradient-based structure, which turned into sensing-only mode (Figure 2.4(c)). By this way, we managed to take into account application requirements (i.e., radio density) while using routing information to reach low number of fully active devices. LPL configuration of each node can further be deduced from both its layer and activity status.

2.3 Main results

The following results were obtained either from simulations or real experimentations. We aimed at validating the identification of sensing-only nodes for the purpose of heterogeneous LPL configurations. We first evaluated the performances of homogeneously configured sensor networks. We then analyzed the proportions of nodes that could be turned into sensing-only mode, based on either LMST or gradient construction, as introduced in Sections 2.2.1 and 2.2.2 respectively. We then provide some results about our initial experimentations of activity scheduling techniques (Section 2.2.3) before detailing the results achieved by our sleep depth approach (Section 2.2.4).

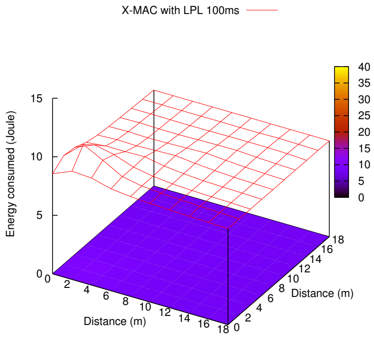
2.3.1 Evaluation of homogeneously configured sensor networks

In order to shed some light on the issues that we outlined in Section 2.1, we simulated a large grid network of hundred (10x10) wireless sensor nodes [3]. The sink was located at the bottom left corner of the grid. All sensors first operated in a event-driven manner, thus not sending any data as long as they had not caught any event. Random geographic routing had been used while the energy model had been configured from the consumption of a Chipcon CC1100 chipset [4]. The performances of X-MAC with preamble length varying from $100ms$ to $500ms$ (all nodes using the same pre-configured preamble length throughout the simulation) were not surprising as they confirmed the wastage induced by homogeneously configured sensor nodes.

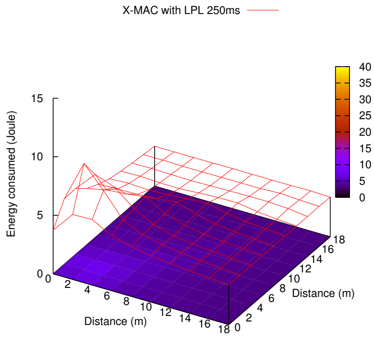
As observed on Figures 2.5, the shorter preambles, the more uniform the energy consumption. The length of the preamble has a direct impact on several energy-consuming factors. First, the preamble length appears as the main factor of energy consumption when a packet is transmitted or received, even though X-MAC limits the energy spent in reception (i.e., early acknowledgments to interrupt the preamble or radio switching off upon preamble destined to other node(s)). Second, the sampling period on each node was the same as the preamble length. The smaller the preamble, the more often nodes sampled the medium, and thus spent energy. Conversely, long preambles induced high costs in places where nodes forward more packets (typically around the sink) but greatly decreased in other places, as channel sampling is performed less often. No ideal LPL configuration could be defined when confronting preamble sampling protocols to a dynamic scenario at large scale. Small preambles reduced communication durations and hence saved energy on the routing path, but required more regular channel sampling that induced energy consumption. Accordingly, the node-to-sink delay increased with higher LPL values, which could provoke congestions around the sink and emphasized the funneling effect. Those results advocated for heterogenous LPL values over the set of nodes.

[3] WSNet, an event-driven simulator for large scale wireless sensor networks. `http://wsnet.gforge.inria.fr`

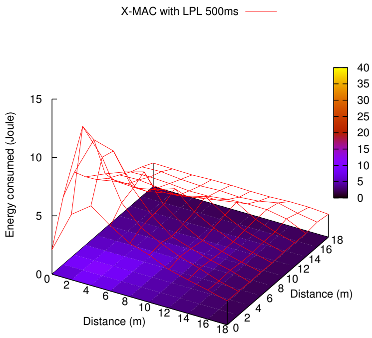
[4] Chipcon CC1100 chipset. `http://ti.com/lit/gpn/cc1100`



(a) $100ms$ X-MAC.



(b) $250ms$ X-MAC.



(c) $500ms$ X-MAC.

Figure 2.5: Energy consumption in grids of devices homogeneously configured with X-MAC.

2.3.2 Achieved proportions of sensing-only nodes

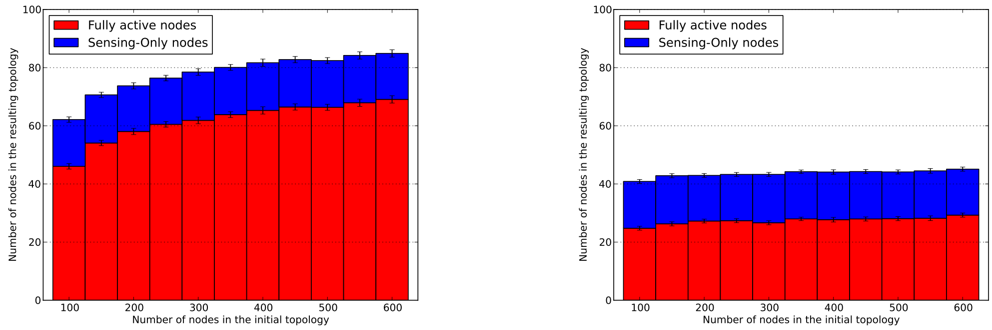
We first applied our PO and PR activity scheduling mechanisms[5], before further identifying sensing-only nodes with a LMST or gradient construction.

Simulations were performed using the WSNet event-driven simulator [71], with networks of 300 to 1000 static sensor nodes, deployed according to a Poisson point process of intensity $\lambda > 0$ over a $50m \times 50m$ square area. Our results were obtained by simulating realistic conditions such as X-MAC [72] layer, log-distance pathloss propagation and orthogonal interferences.

Determination of a sensing-only subset of nodes using a LMST

At first, we evaluated the performances of a LMST whose leaves were identified as sensing-only nodes, as described in Section 2.2.1. Figure 2.6 displays the proportion of fully active and sensing-only nodes observed while using both activity scheduling mechanism and the LMST construction.

[5] A node u is said to be covered as soon as every point of its sensing area (modeled as a disk of $10m$ radius) can be sensed by at least one active sensor.



(a) positive-only mechanism

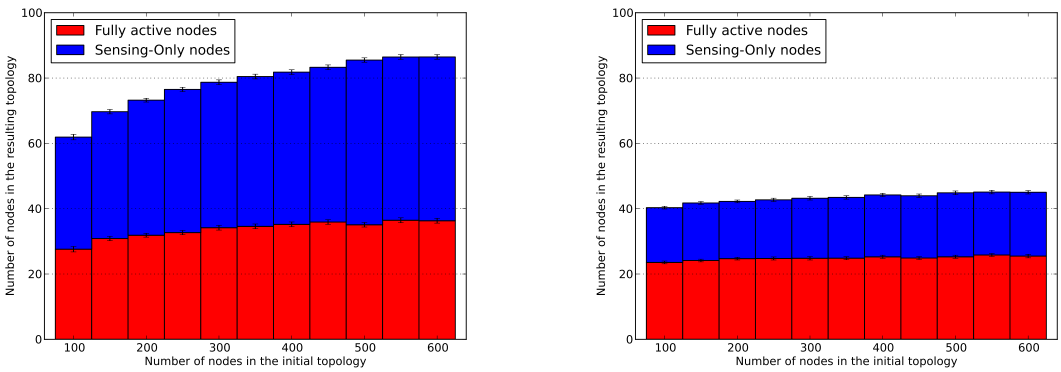
(b) positive-and-retreat mechanism

Figure 2.6: Determination of sensing-only set using a LMST.

As depicted in Figure 2.6(a), approximatively a fifth of active nodes enter sensing-only state while using positive-only. Similarly, while using positive-and-retreat and LMST (see Figure 2.6(b)), a third of the nodes composing the initial topology are set to sensing-only state.

Determination of a sensing-only subset of nodes using a gradient construction

Similarly, we evaluated the performances of a gradient construction on the proportion of leaf nodes in the network. Figure 2.7 shows the proportions of fully active and sensing-only nodes in the network, when using a gradient construction. the gradient enables more than half the actives nodes of the network to be placed in sensing-only mode when coupled with positive-only (see Figure 2.7(a)) and more than a third when coupled with positive-and-retreat (see Figure 2.7(b)).



(a) With positive-only mechanism

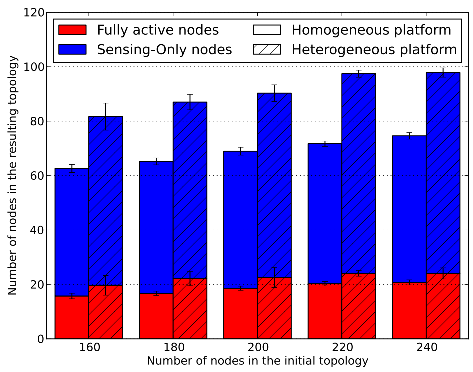
(b) With positive-and-retreat mechanism

Figure 2.7: Determination of sensing-only set using a Gradient construction.

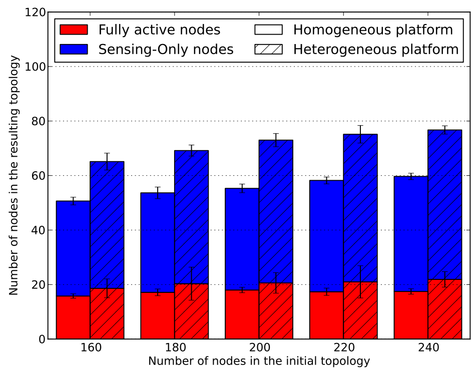
The gradient structure, which only requires an average of one message per active node and no prior neighborhood knowledge, displayed the best performances in simulations. We thus decided to conduct a real large-scale experiment as detailed in the following section.

2.3.3 Experimental assessment of activity scheduling techniques

When Julien Beaudaux launched these experiments, not only did he put up with a number of early problems due to the very recent availability of large-scale testbeds, but also he conducted the first real experimentation of activity scheduling mechanisms. This campaign was performed over the large-scale SensLAB open testbed [8], that would further become part of the FIT IoT-lab facility [73]. We used two platforms, one being homogeneous (a $10m \times 8m \times 3m$ 3D grid, located in Strasbourg, France), and the other heterogeneous (sensors being placed at random positions over a $16m \times 14m \times 3m$ area, located in Grenoble, France). The following results are only based on topologies where the connectivity could be guaranteed.



(a) Positive-Only and a gradient construction



(b) Positive-Retreat and a gradient construction

Figure 2.8: Activity scheduling and sensing-only determination.

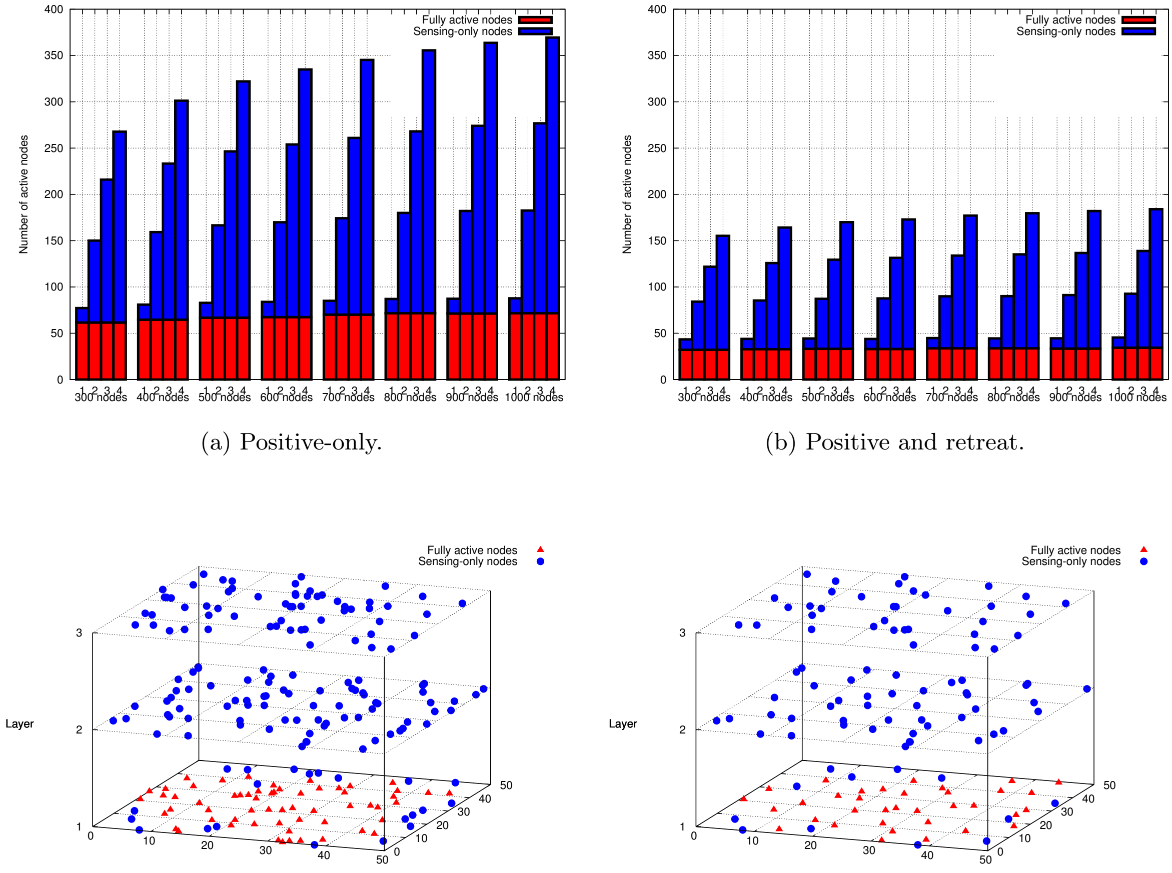
Figure 2.8 depicts the proportion of active nodes in the resulting topology, after application of our positive-only and positive-and-retreat mechanisms. Among the remaining active sensor nodes, our gradient construction allowed at least two thirds of them to enter sensing-only state.

These results confirmed the feasibility of both activity scheduling and sensing-only identification in real radio conditions. We could conclude that large sets of sensing-only devices could be constituted, thus validating our objective to use application requirements as a direct input for MAC configuration. LPL values were later determined based on the activity state of each node, as detailed in our sleep depth approach.

2.3.4 Performances of our sleep depth approach

Our sleep-depth approach cas evaluated over the large-scale SensLAB open testbed (Strasbourg platform). We implemented our contribution in conjunction to the preamble-sampling protocol X-MAC and deployed it over connected multihop topologies. Nodes had to select LPL sampling period (and preamble length accordingly) ranging from $125ms$ to $500ms$.

As observed on Figure 2.9, upon combining a gradient construction and a layered virtual architecture to increase the number of sensing-only nodes, $\sim 80\%$ of nodes became sensing-only (for 300 deployed nodes and required 4-coverage). Furthermore, we could observe that PO combined with routing information allowed several sensors to turn into sensing-only state, all of them being logically located next to the borders of the target area, as confirmed by Figures 2.9(c) and 2.9(d).



(a) Positive-only.

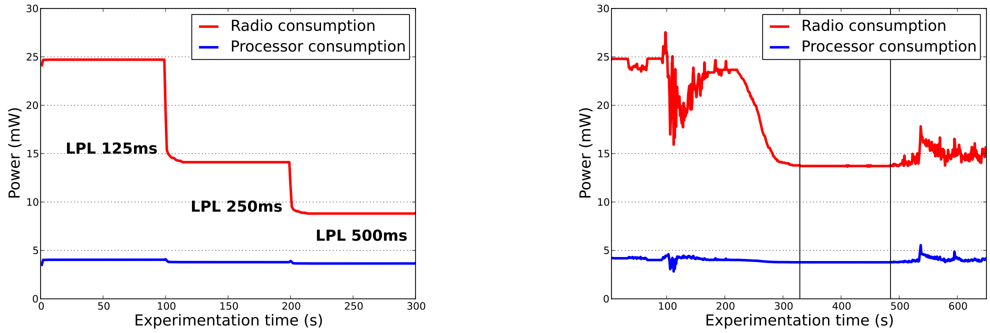
(b) Positive and retreat.

(c) Positive-only (300 node topology).

(d) Positive and retreat (300 node topology).

Figure 2.9: Combined sleep-depth and gradient-based sensing-only identification.

We then evaluated the energy consumption of a free from traffic WSN, with different LPL values (Figure 2.10(a)). All nodes had been initialized with sampling periods of $125ms$, which were then doubled every $100s$. Consumption of sensor components varying greatly from one to another, we did not take it into consideration in this study. We could confirm that the microprocessor was responsible for less than $5mW$ of the average nodes energy wastage, while the radio component represented between $5mW$ and $20mW$. We could observe the importance of LPL values as switching from $125ms$ to $250ms$ was reducing the energy consumed by 40%, and even by 65% when changing from $125ms$ to $500ms$. As illustrated on Figure 2.10(b), the consumption of a network using a sleep-depth was equivalent to all nodes being assigned a $125ms$ sampling value. Indeed, the first period ($0ms$ - $334ms$) corresponds to the initialization of our virtual layers after which all nodes are able to self-determine their LPL configuration. We maintained the network traffic-free between $334ms$ and $493ms$ before introducing homogeneous data transmissions from all nodes.



(a) Three distinct homogenous LPL configurations.

(b) Sleep depth based LPL configuration.

Figure 2.10: Energy consumption.

2.4 Conclusion and future works

In this chapter, we presented several techniques to enable heterogeneous LPL configurations throughout a given network. We took advantage of various information available at each node (i.e., location, application and routing role), thus providing localized solutions with limited control overhead (i.e., at most one control message per node in the network). We managed to identify large sets of sensing-only nodes, that kept participating in the monitoring application while not being involved in multihop communications, thus adopting appropriate LPL configurations (i.e., longer sampling periods). We studied various techniques, based either on the sole physical topology of the network (i.e., LMST construction), the routing tree (i.e., gradient-based) or the application requirements (i.e., connectivity, coverage). Our experimentation campaigns highlighted that our proposals for automatic heterogeneous LPL configuration could offer a better compromise between energy efficiency and loss rate than with any homogeneous setup.

These contributions were all proposed during Julien Beaudaux's PhD and were published in several conferences and journals [3, 4, 6, 7].

Multiple perspectives emerged from those contributions. First, as the application requirements and routing topology would evolve over time, the proposed solutions should consider regular updates in layer or activity status, thus allowing the network topology to remain consistent to changes made at upper layers. Furthermore, several network metrics (e.g., traffic, link quality) could be passively monitored in order to trigger duty-cycle changes only when exceeding a certain threshold. This would help our layer 2 contributions handle the necessary backup nodes, interfaces and links upon failure. Eventually, any application or network input could impact the LPL configuration. For instance, the LPL values would be temporarily adapted, to handle traffic changes, as detailed in Chapter 3.

Chapter 3

Auto-adaptation of low-power listening

Contents

3.1 Research context

Pre-configuration of LPL based protocols allows for various scenarios to be considered. Their performances can be guaranteed as long as initial assumptions remain valid (e.g., network topology, data origins, traffic load). In order to remain energy-efficient over various deployment characteristics, some preamble-sampling protocols expose interfaces to the upper layers to ease the pre-configuration of LPL parameters. As discussed throughout Chapter 2, some of these so-called versatile MAC layers provide sample configurations for particular applications (e.g., B-MAC [43] or X-MAC [72]). This possibility to pre-configure the LPL prior to the deployment used to make it a suitable solution over multiple application characteristics (such as periodic monitoring [74] or sporadic reports [53]), yet demanding that the network topology and the associated data traffic could be anticipated.

Actually, more and more contemplated scenarios were making both the network topology and the data collection scheme dynamically evolve with time while being unpredictable. For instance, assuming a large sensor network that initially operates in an event-driven manner, nodes located around an event of interest would start reporting information in a time-driven manner toward sink stations, at a certain rate. Such events happen in unpredictable locations (e.g., target detections [75], seismic vibrations [76]) thus preventing from any pre-configuration of surrounding nodes. Moreover, most deployments rely on a sink-rooted tree for the routing [52, 77]). Thus, only a subset of deployed nodes participate in relaying the reported data toward sink stations, as earlier discussed (Section 2.2.2). This also leads to high variations in the traffic load that each device has to handle.

Obviously, preamble-sampling protocols lack an optimal configuration for such antagonist traffic patterns throughout the network. Therefore, we studied to what extent versatility could apply to such dynamic scenarios. On-the-fly reconfiguration was required to ensure energy-efficiency at the MAC layer, while addressing scenarios with time-varying or spatially non-uniform traffic loads. More specifically, we aimed at adjusting preamble length and sampling periods in order to adapt to the network conditions, while ensuring fair performances in the whole network.

We had initially emphasized that the main difficulty with such adaptations was to preserve network connectivity, while allowing each node to compute its LPL parameters in a localized manner [1]. We then favored localized solutions for their significantly lower communication overhead and assumed scalability. Still, in order to prevent any node isolation, auto-adaptive algorithms that adjust the LPL values had to guarantee that the preamble length (later abbreviated as PL) of the sender would match the sampling period (noted as SP) of the receiver.

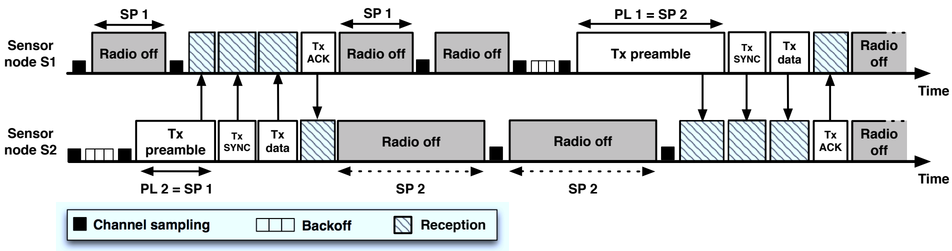


Figure 3.1: Guaranteed communication with uncorrelated LPL values.

Figure 3.1 depicts a node $S1$ using a PL longer than its SP, in order to reach $S2$, whose SP is longer. Conversely, $S2$ is allowed to use a PL smaller than its SP to communicate with $S1$. Communication is guaranteed as long as the sender's PL matches the receiver's SP.

Furthermore, the solution had to be resilient to message losses, while ensuring a low computation complexity. Over the 2008-2012 period, we investigated numerous on-the-fly adaptations of LPL parameters. Overall, a framework for runtime adaptation of MAC parameters had been investigated in [66], using a collection of layer-dependent and independent metrics. This contribution demonstrated that MAC parameters adaptation is both realistic and advantageous for IoT networks. Yet, a dissemination of all MAC parameters throughout the network was required hereafter, while MAC parameters were being kept identical for all nodes.

Control theory was investigated to adapt the SP and PL dynamically at each node [78]. Each sensor node could compute these values independently from its neighbors, without any required control message. However, only single-hop topologies were targeted and packet losses occurred among communication endpoints with different values. The same authors further proposed to dynamically decide of the MAC protocol (i.e., X-MAC, MX-MAC and SpeckMAC), based on some parameters (e.g., packet size) [79]. The receiver was sensing the channel regularly, without any knowledge of the protocol being used by the sender. Auto-adaptation of the LPL values within each protocol were however not discussed.

Other approaches relied on either traffic optimization or observation to increase medium utilization or deduce LPL values respectively. For instance, the payload was appended directly to the preamble strobes, and multiple consecutive frames destined to the same node were aggregated into a single packet [80]. Some traffic indicators also allowed the receiver to double its duty-cycle in case of more packets to come, thus avoiding too long preambles on the sender side. The Low-

Latency Asynchronous MAC (LA-MAC) protocol also allowed nodes to adapt their behavior to varying network conditions, by using network dynamics and priority of requests (e.g., the number of active users, age of a burst, burst size, instantaneous traffic load) [81]. LA-MAC was strongly depending on the routing protocol being used since neighbors were organized accordingly (e.g., DAG, Clustered Tree, mesh). LA-MAC could achieve low latency and energy consumption, but induced an overhead in terms of construction and maintenance of the structure. Similarly, AADCC (Asymmetric Additive Duty Cycle Control) aimed at smoothly reacting to an increase or decrease of channel contention [82]. The number of consecutive packet transmissions was being used to either increment or decrement the sleep time. When the rate of incoming packets reaches predefined threshold values, MaxMAC allowed each node to change their duty-cycle by allocating so-called Extra Wake-Ups [83]. Those duty slots were unallocated once the rate was dropping below the threshold.

Solutions	**Traffic-dependent**	**Decentralized**	**Heterogeneous**
pTunes [66]	✓	×	×
MX-MAC [78], MiX-MAC [79]	×	✓	×
BEAM-MAC [80]	×	✓	✓
LA-MAC [81]	✓	×	✓
AADCC [82], MaxMAC [83]	✓	✓	✓

Table 3.1: Summary of state-of-the-art contributions.

Globally, among the state-of-the-art solutions that were addressing dynamic and bursty traffic, most of them were using excessive numbers of control messages (e.g., parameter negotiation, collection of statistics) while some adopted either a centralized computation or homogeneous configurations, whose efficiency and scalability were yet to demonstrated. As summarized in Table 3.1, solutions such as AADCC or MaxMAC thus appeared as the most relevant in our targeted context, being both adaptive and traffic-aware. We therefore selected these solutions as best candidates for further comparisons during our evaluation campaigns.

3.2 From versatility to auto-adaptation

While interfaces available in a MAC layer enable versatility, we aimed at auto-adapting to the current network traffic, while using various preamble-sampling MAC layers, and working for every traffic pattern. We thus used the available versatility features (e.g., as provided by X-MAC) to automatically tune the LPL mechanism and optimize energy savings.

3.2.1 Implicit auto-adaption to traffic load

We first introduced BOX-MAC, an auto-adaptive algorithm based on a link-by-link implicit agreement in order to establish energy-efficient communications over certain paths towards the sink station [9, 10]. Our goal was to enhance data communication paths along which each relaying node adopted a short preamble thus reducing energy spent during transmissions. Nodes not involved in any communication paths kept sleeping for long periods of time, thus saving energy. The paths leading from the data sources to the sink station could then be composed of energy-efficient links, noted as EE links. The two different durations for PL and SP were assumed to be fixed prior to the deployment and noted as T_{max} for the long one, and T_{min} for the short one. As opposed to B-MAC and X-MAC, one characteristic of our solution was that PL and SP could take different values on the same node. PL was managed under the sending mode while SP was configured in reception mode. Two nodes u and v configured with different PL and SP could not be desynchronized. Compliant with any other preamble-sampling MAC protocol, we particularly studied its association with X-MAC, and observed that energy savings were not accomplished to the cost of reliability.

Operations on the sender side

First, both ends of a EE link must agree on a short preamble ($PL = T_{min}$). The negotiation is initiated by data transmissions. Senders aim at reducing the size of their preambles in order to save energy during further transmissions. By this way, upon reception, each receiver implicitly knows that the sender wants to reach a short preamble.

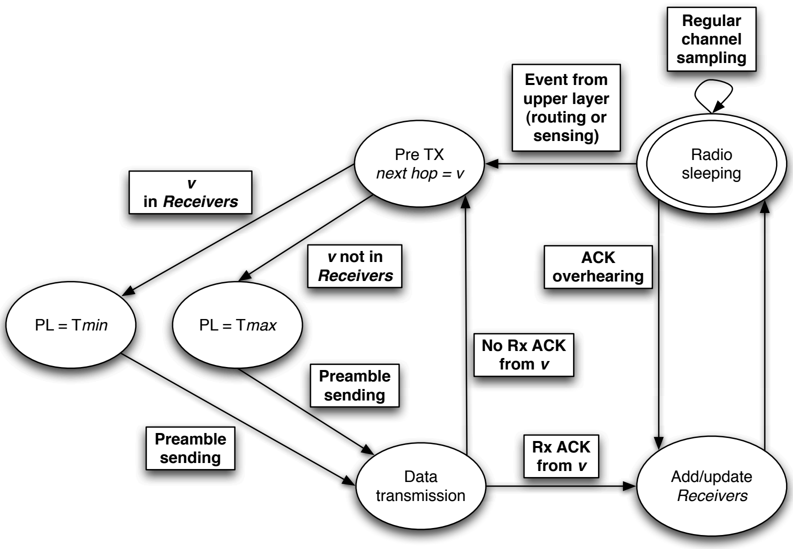


Figure 3.2: State machine of a sender.

Data acknowledgment were considered by the sender as an agreement regarding the link on which next transmissions would require an energy-efficient short preamble only. Agreements have a validity period. Every sender u keeps an up-to-date list of the receivers (noted as $Receivers_u$) that have agreed on a short preamble. For any transmission to node v, u checks $Receivers_u$ and, if an entry exists for v, u adopts a short preamble (i.e., having configured a short sampling period accordingly). Figure 3.2 depicts the state machine of a sender, where *No Rx ACK* refers to the situation where v could not receive correctly the data, or the acknowledgment sent back by v was lost. We also took benefit of overhearing of acknowledgments. For every acknowledgment sent by a node v to a node w, and overheard by a node u, u can decide to add v to $Receivers_u$. By this way, if u sends data to v before the end of the associated timeout, a short preamble can be used. Note that v does not need to know about the adoption of a short preamble by u: the acknowledgment notifies that v has switched to a short SP, hence it will be able to synchronize with the short PL of u.

Operations on the receiver side

The procedure is detailed through Figure 3.3. Receivers do not have to maintain a list of senders. Let a node v configured with $SP = T_{max}$. The first time a node u sends data to v, v acknowledges the reception (noted as *Data OK* in the figure) and agrees on a short preamble, thus changing SP to T_{min}. The time period during which v will remain in this state is noted as $Timeout_v$ (its value is the same as the one used for the timeouts on the sender side). Every new data reception resets the ongoing timeout to $Timeout_v$, thus extending the period during which v keeps an SP equal to T_{min}. In case v receives data destined to another sensor node or if the data is incorrect or incomplete (noted as *Data error* in the figure), it does not send any acknowledgment and goes back to the state it initially came from.

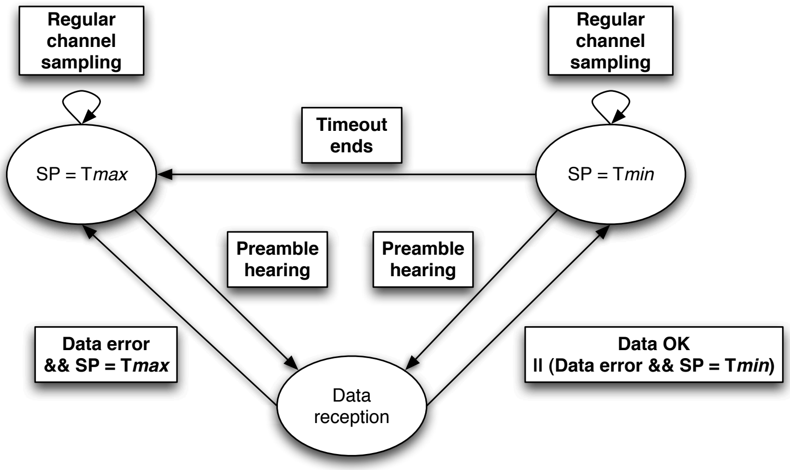


Figure 3.3: State machine of a receiver.

Overall operation

A general overview of our proposal is depicted in Figure 3.4. The sender only modifies its PL value upon an acknowledgment received from the forwarder. It will then be able to send the next data packet using a short PL. The forwarder modifies its SP value to the minimum one upon reception of the first data packet from the sender node.When forwarding the first packet, it however still uses a long preamble. It will reduce the preamble size only from the second forwarded packet (not displayed in the figure). The receiver receives the first forwarded packet, acknowledges it and switches its SP value in order to be ready to hear the second packet on time. According to its role (sender only, forwarder, receiver only), a node modifies either its PL value, or its SP value, or both.

This solution is also resistant to loss of messages that inevitably occur in real radio environments. If data packets are not received, then retransmissions occur until the limit is reached, leading to packet drop. If acknowledgments are lost, the receiver has changed its SP to T_{min} while the sender keeps on sending a preamble of length T_{max}. Such situations can not lead to link-layer disconnections: the sampling period being shorter than the preamble, the receiver will inevitably hear the preamble. It simply wastes some energy that could have been saved otherwise.

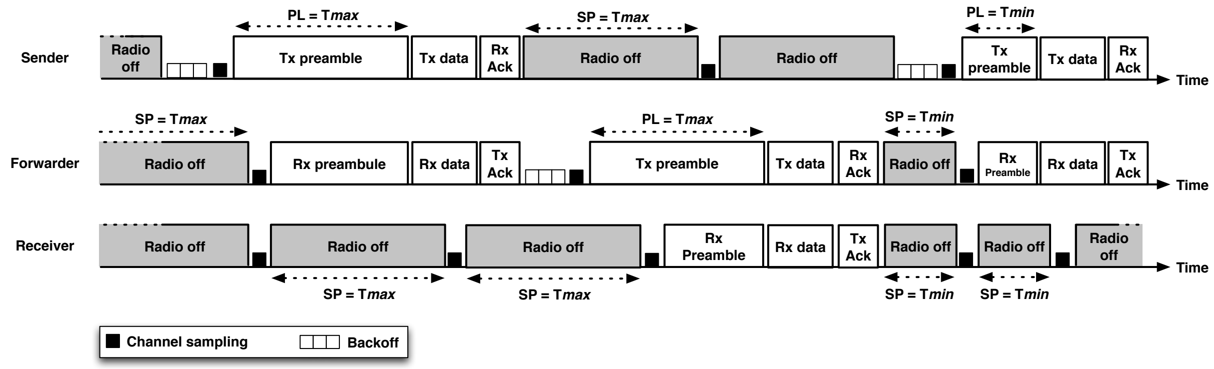
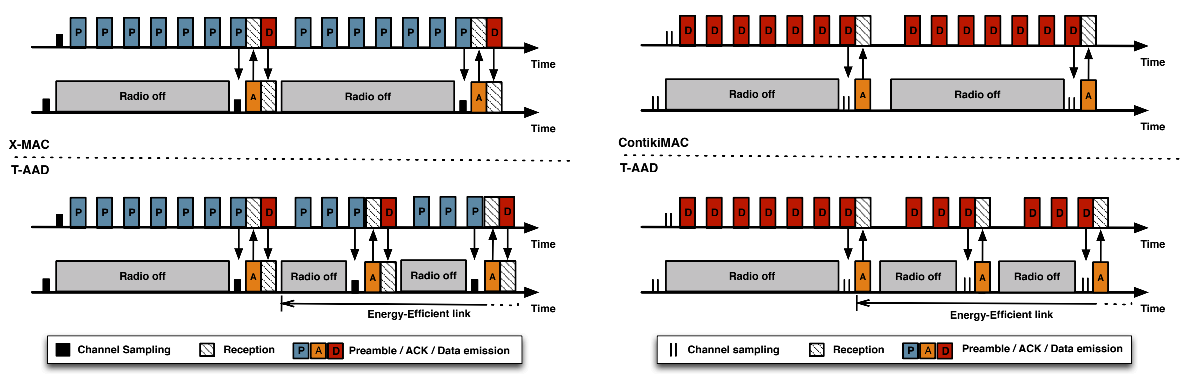


Figure 3.4: Overview of the solution with one sender, one forwarder and one receiver.

3.2.2 Assisted auto-adaption with anticipated traffic load

We further worked toward a traffic auto-adaptive mechanism able to anticipate traffic load changes by tuning the MAC parameters according to the upcoming traffic volume of concerned nodes [11]. In our T-AAD solution, nodes had to decide autonomously of their own parameter settings, thereby trying both to increase the network lifetime (i.e. reduce to minimum the energy consumption) and to decrease the latency. We extended our initial proposition of auto-adaptation at runtime by allowing nodes to announce the anticipated volume of upcoming traffic, thus turning our mechanism into an explicit negotiation between senders and receivers. Traffic sources and relays will now assist receivers to determine the most appropriate LPL values.

In particular, we tuned the duty-cycle based on the amount of data packets that a node is expected to transmit (i.e., through MAC or application queue). While nodes running BOX-MAC had a predefined timeout for energy-efficient links (i.e., switching back to initial LPL values upon too long silence from the node of the other end), we proposed to calculate this timeout based on the inputs received from the sending node. A receiving node should switch from long to short sampling period (SP) for the time of the traffic variation, and then back to long again. Figure 3.5 illustrates T-AAD, as applied to both X-MAC and ContikiMAC protocols.



(a) Applied to X-MAC. (b) Applied to ContikiMAC.

Figure 3.5: T-AAD: lightweight traffic auto-adaptations for low-power MAC protocols.

In each packet, one additional piece of information indicates the total number of packets that a node intends to send in a row. This value is used to calculate the time (here referred as T_{adapt}) that a node will maintain a short sampling period. Hence, all nodes start with identical SP values (e.g., $500ms$, $1000ms$, $2000ms$), depending of the initial user deployment configuration. Once a traffic load variation occurs, the receiving node calculates T_{adapt} once receiving the first packet. Then, upon successful reception, it switches its SP to minimum, depending on the hardware (e.g., $32ms$, $64ms$, $125ms$). Once the T_{adapt} period ends, the receiver switches back to a regular channel sampling state. T-AAD allows nodes to save energy when using a long SP, while achieving shorter delays when switching to short SP.

T_{adapt}, the actual time that a node will stay in short SP, is a one parameter function (i.e., number of packets). Let SP_{max} be the time period for long SP and SP_{min} for short SP respectively. Q_{len} designates the total number of packets that a transmitter is expected to send, and finally let M_{err} be the margin of error. We considered the margin of error due to the probability of having packet retransmissions, consequently to packet losses. This situation can be caused by collisions or interferences in the network. Hence, in order to ensure that the calculated period is at least as long as the traffic load change (including MAC retransmissions), a margin of error is added depending on deployment conditions (e.g., loss-rate, perturbations). Hence, this observation can be modeled and computed as follows:

$$T_{adapt} = SP_{max} + (Q_{len} - 2) \times SP_{min} \times (1 + M_{err}) \tag{3.1}$$

Let us assume that a node A has n packets to be transmitted towards a node B. The queue length Q_{len} equals to n. There are $n-1$ sleeping periods among the n consecutive packets. The receiver is in SP_{max} mode for the period between the first two consecutive packets, due to the time required by the hardware to update the SP. The remaining $n-2$ sleeping periods will be in short SP. Thus, we are adding SP_{max} in the beginning of our equation and $n-2$ times the SP_{min}. Hence, since each packet contains the information about the number of packets in the queue (i.e., Q_{len}), node B will update T_{adapt} and switch to SP_{min} for the exact estimated time.

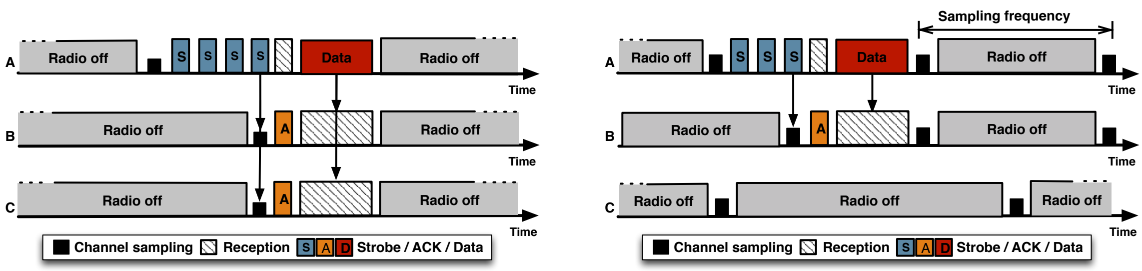
3.2.3 Mitigation of packet duplications

In this chapter, we have presented several techniques to allow for heterogeneous configurations of LPL-based MAC protocols. As an additional service provided by the MAC sub-layer to upper layers (e.g., routing), we considered the opportunity to use heterogeneity as a mean to limit the propagation of non-necessary packets throughout the network [12].

For each communication, the routing protocol identifies the most suitable next-hop for the intended destination. Several deployments opt for minimal control overhead by using opportunistic routing, where packets are sent in anycast. Hence, a data packet is transmitted to the first potential forwarder (e.g., any neighbor closer to the sink in term of hops) that acknowledges the corresponding message [84].

By increasing the number of potential forwarders, an opportunistic routing process has a straight impact on MAC layer performance. Indeed, several neighbors competing for the same packet transmission necessarily lead to faster (yet multiple) preamble acknowledgements. As a result, opportunistic routing allows for improved throughput, reduced end-to-end delays and more balanced energy consumption between nodes in the network. It also increases resiliency to network dynamics [85].

However, in WSNs operating with preamble-sampling MAC protocols under homogenous configurations, a single packet could be received by multiple neighbors. Indeed, two or more potential forwarders may sample the radio channel at the same time, thus simultaneously detecting and acknowledging a strobe associated to the ensuing packet transmission. The transmitter would most probably receive one of those acknowledgements and directly transmit the data packet. Yet, multiple receptions will be triggered by all neighboring nodes that have acknowledged the preamble.



(a) A single packet may be received by several nodes sharing the same wake-up slot.

(b) Heterogeneous LPL configuration allows some nodes to be deaf to the transmissions.

Figure 3.6: Packet duplication (left) and deafness in heterogeneously configured network (right).

This phenomenon leads to several distinct forwarded messages, recursively propagated at each hop, and eventually to packet duplication at the sink. In parallel, overall network performance is affected due to the corresponding traffic and channel occupancy increase.

The probability of two or more nodes simultaneously sampling the medium for incoming packet is strongly correlated to the total number of potential forwarders along with their wake-up interval, as exposed in [86]. In this paper, Landsiedel *et al* also presented a complete opportunistic routing solution in which a double acknowledgement mechanism was introduced to solve the unique forwarder problem. The selected unique forwarder is thus the first among the potential forwarders to acknowledge the preamble, but also the one whose second acknowledgement was the faster (after a random timeout). Other contributions have tried to mitigate packet duplications. Among them, authors of [87] filtered duplicates at the routing layer. Although reducing unnecessary forwarding, This solution solves the problem only partly, as duplicates are identified only upon using the same joint paths.

From the MAC layer standpoint, two factors need to be fulfilled for a single data packet to be received by several neighbors. First, two or more eligible nodes have to sample their radio channel while the packet is being transmitted (Figure 3.6(a)). Depending on the channel sampling rate and network local density, the probability for this condition to be met varies. Indeed, the more frequently nodes sample their radio channel (and thus the higher its duty-cycle), the more likely several of those are to be awake during the packet transmission.

We formalized this problem as a birthday paradox. Each node wakes its radio up to sample the medium for incoming messages (sent in anycast) under regular intervals. This channel sampling procedure takes about $7ms$ on typical radio chipsets. The probability of two potential forwarders to sample the medium at the same time slot (and thus catch the preamble and acknowledge it) is given by Equation (3.2), with the channel sampling duration called *sampling duration* (e.g., $7ms$).

$$P_1(X) = \frac{1}{\frac{wake-up\ interval}{sampling\ duration}} = \frac{sampling\ duration}{wake-up\ interval} \tag{3.2}$$

Instead of looking for any pair of nodes sharing the same wake-up slot as with the birthday paradox, we here considered that a node had already caught the strobe, and then evaluated how many other potential forwarders would share the same wake-up slot. By utilizing the number n of potential forwarders, we formalized in Equation (3.3) the probability of having two or more of those catching the preamble, and thus getting the data at the same time slot. To do so, we first formalized its complementary: considering one first acknowledging node, the probability that no other potential forwarder samples its channel at the same time slot is given as follows:

$$P_2(X) = 1 - \overline{P}(X) = 1 - (\prod^{n} P_1(X)) \tag{3.3}$$

Figure 3.7 uses equation (3.3) to represent the analytic probability of a single packet to be simultaneously received by multiple nodes. As shown here, this probability is strongly correlated to both the wake-up interval provided to the nodes in the network and the size of the set of potential receivers. Note that, due to the multihop fashion of typical WSNs, this phenomenon may recursively affect forwarding at each hop, thus leading to an exponential packet duplication at the sink.

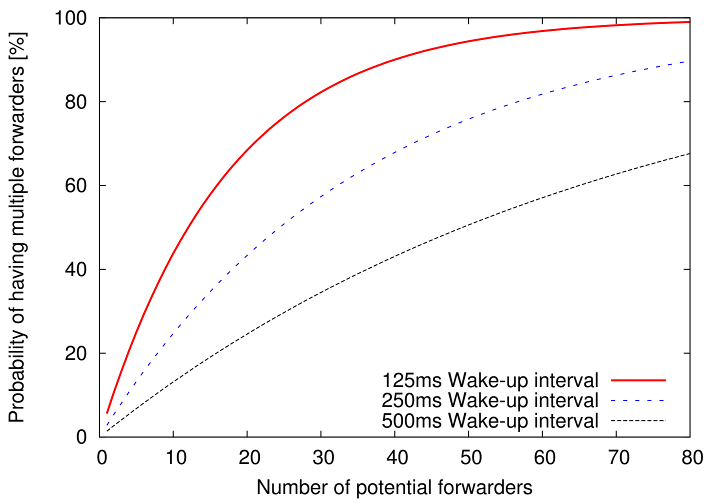


Figure 3.7: Probability that multiple nodes receive the same data packet, depending on the size of the potential forwarders and the wake-up interval.

In this context, we proposed to take advantage of LPL heterogeneity among the nodes in order to have some of them become deaf to selected transmissions. By this way, we aimed at reducing the probability of having multiple receivers for a single packet, by dynamically and automatically adapt the nodes wake-up interval to the local number of potential forwarders (i.e., designated by the routing layer as potential next-hops towards the sink). Once again, our objectives were to enable such selective deafness throughout the network, in a localized manner, and without endangering network disconnection.

We assumed that nodes knew the number of neighboring potential forwarders, by overhearing of control or data messages, through regular broadcast of some hello messages. The number of potential forwarders was used in accordance with application-level performance parameters to affect each node in the network with a specific wake-up interval. Application-level performance parameters can range from expected lifetime of the deployment to Quality of Service (QoS) and resilience requirements (e.g., maximal delay or loss-rate), and depend on the application specificities and requirements [88].

Both application-level performance parameters and the size of the potential forwarders subset were provided as parameters of a multi-objective optimization problem (MOP). The optimal solution to this MOP (or the *Pareto-optimal* solutions) represented the best-suited MAC parameters a node could be configured with to fulfill all application requirements.

Thus, when the set of identified other potential forwarders (i.e., neighbors sharing the same opportunistic routing metric) is larger, the node will select a longer wake-up interval to fit with the lifetime requirement, without exceeding the reliability requirement. Conversely, a reduced set of other identified potential forwarders implied that the node would select a shorter wake-up interval in order to ensure efficient management of incoming messages.

3.3 Main results

The following results were obtained either from simulations or real experimentations. We aimed at showing that our auto-adaptation solutions could operate on top of any preamble-sampling protocol, while performing better than any other homogeneous configuration. We here present the WSNet simulation results obtained with our BOX-MAC proposal (Section 3.2.1). We then detail the performances of our T-AAD solution (Section 3.2.2) before studying the mitigation duplicate packets (Section 3.2.3).

3.3.1 Evaluation of our BOX-MAC solution

In our WSNet simulations, all sensors first operated in a event-driven manner, and sent data upon event occurrence only (one packet every second during the duration of the event, which lasted $10s$). We had uniformly distributed the starting time of 180 events over the simulation time (3 hours) according to a Poisson process. Regarding the auto-adaptation, we chose a $Timeout$ value of 10 seconds, a T_{min} equals to $100ms$ and a T_{max} equals to $500ms$. The remaining simulation parameters are exposed in Table 3.2.

Topology	Square grid (20mx20m) of 100 (10x10) fixed sensors
Data sending period	Upon an event: every second during 10s
Data packet/payload size	25 Bytes / 4 Bytes
Routing model	Random geographic routing[6]
MAC model	X-MAC with a preamble of 100, 250 or 500ms
Radio model	Half-duplex, bandwidth: 15 kB/s
Antenna model	Omnidirectional
Radio propagation model	Friis, 868 Mhz, pathloss: 2 (range: 3m)
Modulation model	BPSK
Energy model[7] With a 3V battery	Idle: 1.6mA, Rx: 15 mA, Tx: 16.9 mA, Radio init: 8.2 mA Idle: 4.8mW, Rx: 45 mW, Tx: 50.7 mW, Radio init: 24.6 mW
Number of events	180 events
Duration and number of simulations	3 hours, simulated 20 times for each combination

Table 3.2: Simulation parameters used in our experiments.

Our simulation scenario consisted in a large 10×10 grid network of wireless sensor nodes, two meters separating each device from any of its neighbors (Figure 3.8). The results exposed hereinafter exhibit the benefits of our solution in terms of energy consumption, end-to-end and one-hop delays as well as packet losses.

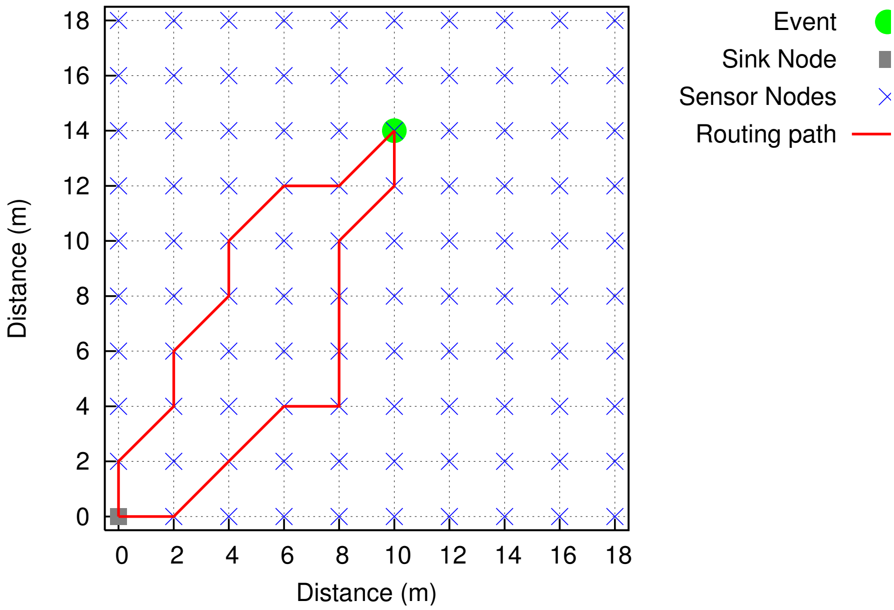


Figure 3.8: The topology used in our simulations. Upon event occurrence, ten packets are generated by sensing devices. Here, two packets (from one given sensor) are routed through different paths towards the sink (random geographic routing).

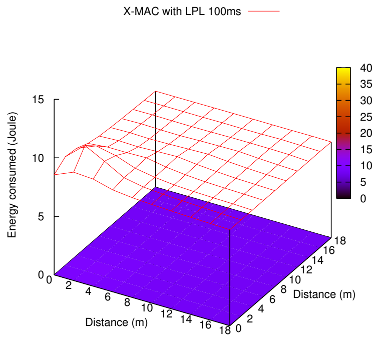
Energy consumption

As detailed in Table 3.3, energy consumption is significantly reduced on the routing path. Network-wide, it uses 31% less energy than when X-MAC is used with a LPL pre-configured to $500ms$ ($261J$ versus $380J$). With our solution, the experiment could last around 1315 days before all nodes would die (against 900 days without our algorithm), and the most consuming node would die after 586 days (against 280 days previously).

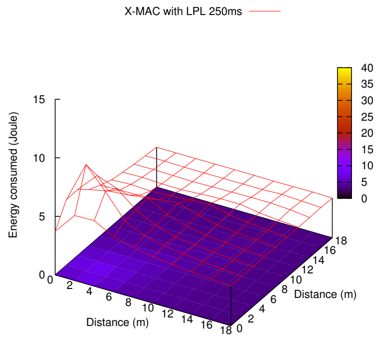
MAC protocol	Total energy consumed in the network (Joules)	Main reasons (% of total consumed)			
		Tx (%)	Rx (%)	Idle (%)	Radio Init (%)
X-MAC (LPL 500ms)	380.48 (± 3.19)	30.62	15.22	19.11	35.04
X-MAC (Auto-adaptive)	261.42 (± 0.98)	14.25	4.57	19.34	61.85

Table 3.3: Average energy spent in the network for X-MAC with the auto-adaptive LPL. For reference, the best values obtained by original X-MAC are also displayed.

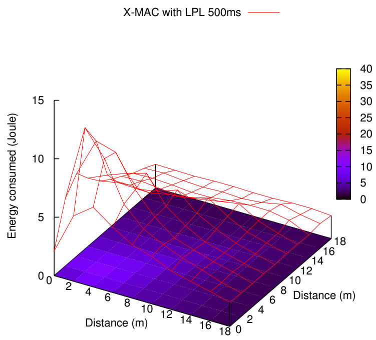
Figures 3.9 depict the maps of the energy consumption for various X-MAC configuration. In order to ease the reading and comparison of auto-adaptive X-MAC (Figure 3.9(d)) with homogeneously configured X-MAC, Figures presented in Section 2.3.1 appear here also (Figures 3.9(a), 3.9(b) and 3.9(c)). We can observe that auto-adaptive X-MAC outperforms any other statically configured X-MAC as energy consumption is maintained at a reasonable level and is more uniformly distributed, despite the use of a geographic routing that could lead to a funneling effect and thus important battery wastage around the sink (e.g., Figure 3.9(c)).



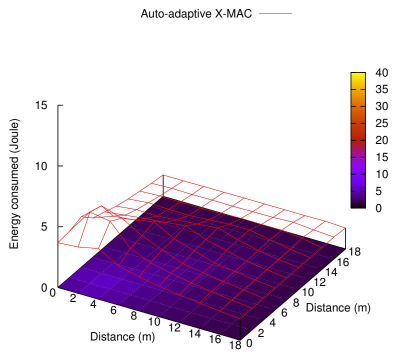
(a) 100ms X-MAC.



(b) 250ms X-MAC.



(c) 500ms X-MAC.



(d) Auto-adapted X-MAC.

Figure 3.9: Energy consumption in grids of devices.

End-to-end and one-hop delays

The average end-to-end delay according to the number of hops to the sink is depicted in Figure 3.10. Auto-adaptive X-MAC achieves roughly the same node-to-sink delay as when it is configured with a fixed LPL value of 250ms. The one-hop delay is 137.6ms ($\pm$ 1.23) in average, and 49% of the preambles sent during the whole experiment are short ones (i.e., initially configured to last a maximum of 100ms).

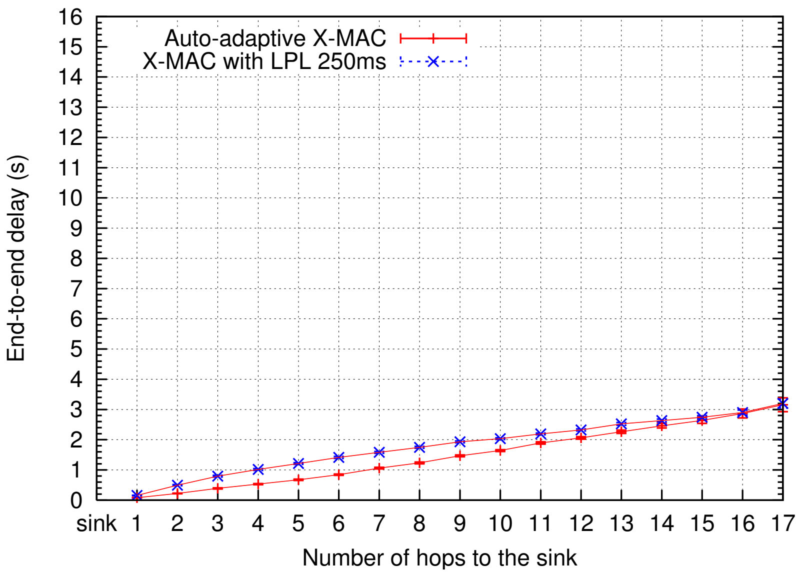


Figure 3.10: End-to-end delay (with 95% confidence interval) according to the number of hops to the sink for auto-adaptive X-MAC. Results for original X-MAC are given as reference.

Packet losses

Losses at the sink are exposed in Table 3.4. Only 4.3% of the data packets did not reach the sink. A retransmission threshold set to three thus allowed to be pretty resilient to losses. Thus, this low percentage of packet losses with heterogeneous configurations has proved that energy savings had not been accomplished to the cost of reliability.

MAC protocol	**Packet sent by applications**	**Packet received at the sink**	**Percentage of losses at the sink**
X-MAC (LPL 250ms)	1800	1691.45 (± 11.56)	6.03
X-MAC (Auto-adaptive)	1800	1722.75 (± 6.51)	4.29

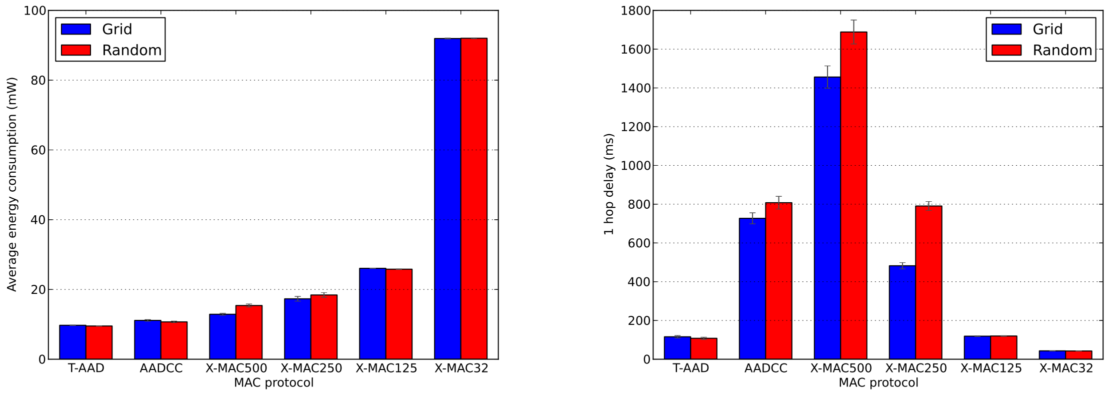
Table 3.4: Average packet losses at the sink (with 95% confidence interval) for auto-adaptive X-MAC. As reference, approaching results obtained by original X-MAC are also displayed.

3.3.2 Simulations and experimentations of T-AAD

After showing that heterogeneous LPL values could defined at each node in a fully decentralized manner, we aimed at pushing those ideas to experimental testbeds where reduced energy consumption, delays and channel occupancy would be observed on real hardware and radio conditions. We thus performed a thorough evaluation of T-AAD, both through simulation (i.e., Cooja simulator [89]) and experimental study over FIT IoT-LAB testbed. In addition, we compared T-AAD both with a statically configured network using X-MAC and state-of-the-art auto-adaptive mechanism such as AADCC [82].

We set $500ms$ for SP_{max} and $32ms$ for SP_{min} respectively as SP configurations for T-AAD and AADCC both for simulation and experimentation. For pre-configured X-MAC, we kept the standard format with SP of 32, 125, 250 and $500ms$ (i.e., X-MAC32, X-MAC125, X-MAC250 and X-MAC500 respectively). At the routing layer, we relied on a broadly used scalable gradient protocol which generated a virtual routing tree rooted at the sink (i.e., using a number of hops towards to the sink as a metric) with small overhead.

During our simulations, 50 wireless sensors were either randomly or uniformly distributed (i.e., grid) in an area of 50×50 meters. The nodes used a time-driven application in which 10 packets were sent every 500 seconds. We compared to a homogeneously pre-configured X-MAC and to a state-of-the-art auto-adaptive protocol AADCC.



(a) Average energy consumption

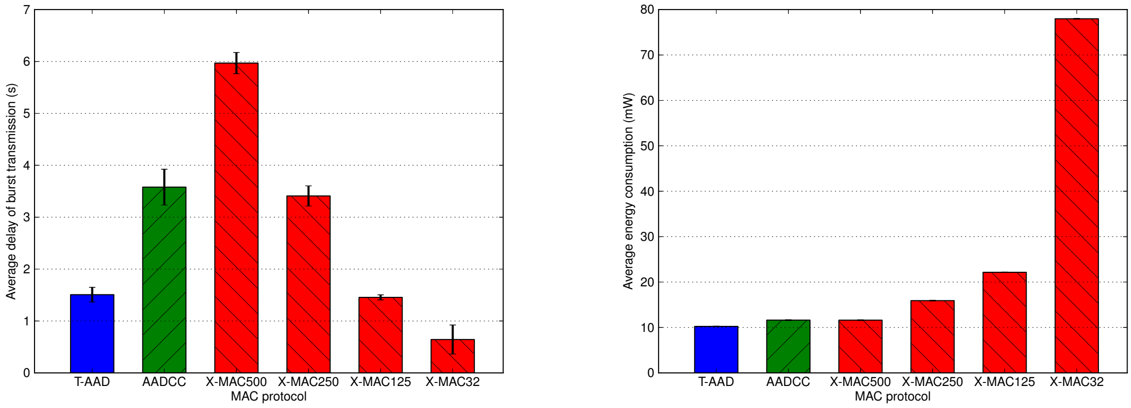
(b) Average one-hop delay

Figure 3.11: Performance of MAC protocols in simulation environment.

As expected, Figure 3.11(a) shows that both T-AAD and AADCC consumed less energy network-wide than any of the pre-configured X-MAC. In particular, T-AAD reduced the energy depletion by about 37% when compared to X-MAC125. We also observed that T-AAD performed slightly better than AADCC. This was mainly due to the long preambles that AADCC necessarily induces for each adaptation process. Regarding the one-hop delay (i.e., including the initial back-off, the channel sampling period, potential congestion back-offs, potential retransmission delay and the preamble length), Figure 3.11(b) highlights the good performances achieved by T-AAD. More specifically, even though T-AAD and AADCC presented similar energy consumption, T-AAD performed almost five times better than AADCC, in terms of delay.

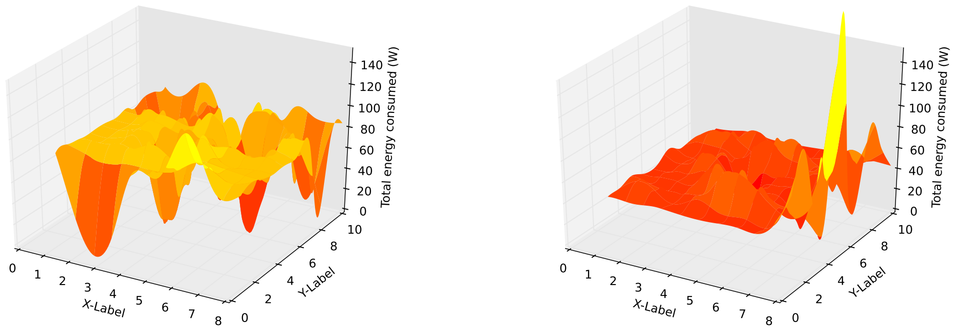
We then ran several experimentations over the open large scale FIT IoT-LAB testbed (Strasbourg's site). In order to limit border effects due to excessive radio reflections, we decided to work with 80 nodes located in the same 2D plan and equipped with a Texas Instruments CC1101 radio chipset and an open MSP430 micro-controller. All 79 nodes were randomly selected as data sources while implementing a time-driven application model (each of them transmitting 10 packets in a row, every 1000 seconds). As per Equation 3.1, the margin of error was fixed at a realistic 15% of the total T_{adapt}. The sink was located at the top left of the area. Each experiment lasted for 70 minutes.

As for simulations, we evaluated the general performances of the network, either configured homogeneously using X-MAC, implementing the AADCC and T-AAD mechanism. Figure 3.12 confirms that T-AAD allowed the network to gain both in delay and energy consumption at the same time as it achieved delays similar to X-MAC125 with an energy consumption lower than those obtained with AADCC or X-MAC500.

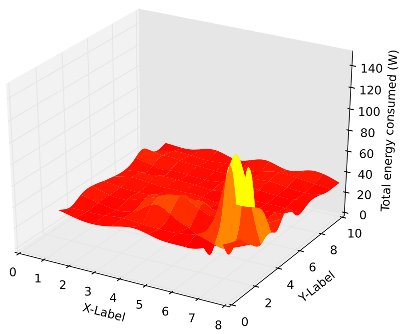


(a) Average delay (b) Average energy consumption

Figure 3.12: Performances of MAC protocols in a complete network setup.



(a) X-MAC125 (b) AADCC



(c) T-AAD

Figure 3.13: Total energy consumption for X-MAC125, AADCC and T-AAD.

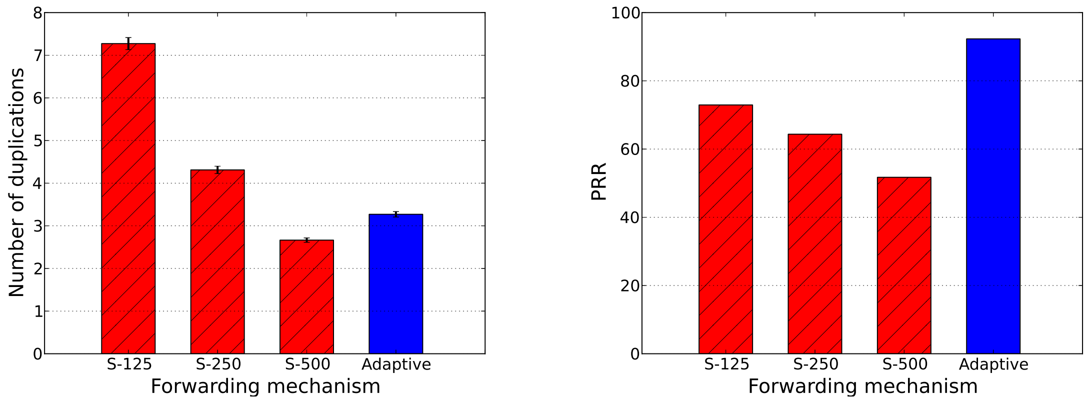
Figure 3.13 illustrates a mapping of the energy depletion throughout the network, in order to identify its repartition and disparity among the nodes. We can observe T-AAD maintained a quite homogeneous energy consumption throughout the network (with the exception of the sink) while X-MAC125 and AADCC vary much, depending on the considered node.

Indeed, with X-MAC and AADCC, the traffic load bears a strong influence on energy consumption (due to long preamble transmissions and the overhearing associated). As a consequence, nodes which had to relay information from their neighbors had to remain active for a larger proportion of time than nodes activated, only to send their own data. T-AAD allowed for a quicker management of packets, so that nodes could go back to sleep after a short period only, thus explaining the homogeneity of its energy depletion.

As a result, T-AAD proved to be more stable and less dependent to the traffic load in the energy consumption of the nodes, as well as providing significative energy consumption reduction when compared to any network homogeneously configured with X-MAC or with an reactive adaptive AADCC mechanism.

3.3.3 Performances of MAC deafness for mitigation of duplicate packets

We implemented our contribution in conjunction with X-MAC protocol and set a time-driven CBR traffic at $1pkt/100sec$, a maximal duplication probability of 60% and no minimal lifetime as applicative requirements. Thus, according to the analysis presented in Section 3.2.3, nodes that had less than ten potential receivers continued with their original MAC configuration (i.e., 125 ms), while the nodes that had more than twenty potential receivers switched to 500 ms. The remaining devices switched to 250 ms.



(a) Average number of duplications at the sink

(b) Packet reception rate

Figure 3.14: The impact of the adaptive mechanism to the network performance.

As illustrated on Figure 3.14(a), the total number of packets in the network (i.e., including both originally transmitted and forwarded packets) logically raised as the wake-up interval was being reduced. This led to a vicious circle where increased congestion, channel occupancy and competition of the medium access enlarged the probability of packet retransmissions and thus the volume of traffic to be handled. MAC deafness helped reduce the multiple receptions of a single packet at the sink (i.e., more than 50% comparing to a static and homogeneous sampling period of 125 ms).

Furthermore, we observed that controlled heterogeneity at the MAC layer could have a benefit impact on end-to-end reliability, in addition to the optimized energy consumption. Almost twelve thousands messages had been transmitted in order to estimate accurate packet reception rate. As depicted on Figure 3.14(b), even though collisions increased when reducing the wake-up interval, the overall end-to-end reliability remained better than statically configured alternatives, whatever the sampling periods. Indeed, by adapting the sampling frequency locally at each sensor node, we reduced the unnecessary traffic and thus the congestion and collisions in the network.

3.4 Conclusion and future works

In this chapter, we detailed two of our main contributions regarding the auto-adaptation of preamble-sampling MAC protocols. The first one, BOX-MAC relied on implicit agreements for energy-efficient links on which short sampling periods and preambles are used on both ends, for an a priori fixed duration. Upon considering traffic anticipation, we proposed a stable, lightweight and traffic independent solution, T-AAD, which outperformed state-of-the-art adaptive solutions such as AADCC. We further examined to what extent such auto-adaptive mechanisms could help mitigate packet duplication induced by the routing layer, by allowing some nodes to be deaf (i.e., long sampling periods to reduce the probability of catching preambles of unnecessary transmissions), according to the local number of potential data relays. We could show that local configurations at each node, made on available information from control-packet transmissions (e.g., routing construction) could leverage the unnecessary overhead, without endangering packet reception rates.

These works were published along with several complementary simulation and experimentation results [9–12]. T-AAD was also demonstrated live at ACM Mobicom 2015, where one transmitter node was changing its traffic load periodically sending from 4, 8, 16 to 32 packets in burst to a single receiver [13].

Several perspectives emerged from these works. First, the automated learning of traffic patterns would allow nodes to determine their configurations with further reduced communication overhead. Second, using routing information is always a sensitive operation, especially when the routing topology is not stable. We should study the impact of layer 2 changes on the operations at third layer in order to work towards a smoother collaboration between both. More specifically, the service provided by layer 2 to the routing layer should allow to hide most of the instabilities induced by the physical layer. Virtual 1-hop neighbors could thus be envisioned in order to allow for more consistent decisions at layer 3. Furthermore, energy-harvesting techniques are now affordable and some can be embedded on small devices that store excessive energy for later use. As discussed in [90], the MAC protocol design paradigm shifts from "how to reduce energy consumption" to "how to optimize performance with harvested energy.". In this context, nodes with limited energy budget should be preserved and granted with collision-free periods while those with higher capabilities might take more risks when trying to gain access to the medium. While some centralized solutions have already been proposed to do so, distributed mechanisms are still to be defined.

Chapter 4

Low-power neighbor discovery in mobile IoT

Contents

4.1 Research context

As discussed throughout the previous Chapter, event-driven networks face occasional, bursty and unanticipated multihop data transmissions. For instance, wildlife monitoring requires limited-memory devices to operate in limited-internet access and immediately upload their stored data at a more powerful device (i.e., sink) upon link establishment [27]. Currently, time-synchronized MAC protocols are not considered for such scenarios, due to scalability and dynamic nature of the network (e.g., mobility). Furthermore, IoT applications have shifted progressively from infrastructure of static nodes to more dynamic topologies where bulk transmission schemes and mobility appear to be essential [14], [91].

While MAC protocols for traffic-aware IoT applications were discussed in Chapter 3, we here focus on how to support mobility for various scenarios (e.g., persons, animals, objects). As of today, most of IoT applications consider static sensors that handle dynamics imposed by radio links (e.g., change of next-hop at routing layer, depending on the physical layer conditions and the status of devices) [68, 92]. Logically, such link fluctuations and disconnections are even more prone to happen as dynamics emerge from nodes movement. Consequently, before addressing dynamics at routing, links themselves must be stabilized, especially regarding mobile nodes and their communications with the static infrastructure.

In order to ease data uploads from low power mobile nodes to any sink-connected neighbor of their vicinity, we focused on establishing links for mobile-to-static communications that further allow for next-hop switching upon data offloading (i.e., handover). We assumed that mobile nodes were not involved in the routing task, thus requiring link establishment and handovers to be performed jointly with the MAC layer mechanisms. We aimed at establishing communication links with selected next-hop static nodes without increasing neither delay nor energy consumption due to potentially longer routing paths.

Classical preamble-sampling MAC solutions failed to meet those requirements, thus, emphasizing neighbor discovery as a key primitive in mobility-aware IoT networks. For instance, ContikiMAC embeds most of innovative features of existing preamble-sampling protocols [60]. In particular, periodic wake-ups have been used by X-MAC [72], the phase-lock optimization has been suggested by WiseMAC [93] and the use of data packet copies as a wake-up strobe has previously been introduced by the TinyOS BoX-MAC [61]. Moreover, it includes bursty transmission handling [94]. However, ContikiMAC was designed for static networked nodes and relies on unicast and broadcast transmissions. Hence, mobile nodes are not aware of next-hop addresses and this actually prevents them from utilizing unicast transmissions. Since broadcasting is a costly alternative, mobile nodes fail to access the medium in an efficient manner. Moreover the increased traffic due to the potential broadcast transmissions degrades network performance.

Eventually, mobile nodes have to switch their static next-hop (e.g., due to movement, link fluctuations, disconnections). Handover procedures must allow mobile nodes to seamlessly change their next-hop in order to maintain uninterrupted communications with the static infrastructure. In this context, random selection (as observed in most mobility-aware MAC protocols, as detailed hereafter) leads to increased end-to-end delay and energy consumption due to potentially instable communications and longer routing paths. This further required to investigate seamless handovers in order not to interrupt a (burst) transmission upon next-hop changes.

In WSN applications such as patient [95] or animal monitoring [27], sensor nodes are attached to persons, animals or objects, while data are delivered from mobile nodes to the sink station over a multihop path, which often consists in a set of static nodes. The whole network is subject to changes, due to mobile nodes that modify the topology. In contrary to the traditional a priori known time-driven traffic patterns, event-driven networks face occasional, bursty and unanticipated multihop data transmissions. Several mobility-aware MAC protocols have been proposed in the literature [96].

In [97], the authors presented MA-MAC (Mobility-Aware Medium Access Control), which extended X-MAC by defining two thresholds for the Received Signal Strength Indicator (RSSI) evaluated on incoming ACK packets from static nodes. The first was triggered upon handover requirement (too long distance detected with current receiver) and led to a new neighbor discovery (broadcasted handover requests), while the second set an upper limit (i.e., distance) for a mobile node to keep its current next-hop, thus allowing for background evaluation of the newly discovered node. MA-MAC introduced communication overhead (new header in the payload part of the packet), while depending on the scheduling of neighboring nodes. It relied on sole RSSI evaluation for the two reported thresholds, thus assuming that the relationship between RSSI and distance is stable, which is fairly critical [98].

Mobility Aware RI-MAC (MARI-MAC) [99] extended the Receiver-Initiated RI-MAC protocol [100] with a distance threshold (based on RSSI evaluation, as in MA-MAC) that would trigger the handover procedure. MARI-MAC assumed that the node mobility model was known a priori. Moreover, the performance of receiver-initiated protocols being highly correlated to the density of senders, the more senders around one receiver, the higher probability of packet collision in the network. As a result latency may increase dramatically.

In [101], [91], Dargie et al. presented MX- MAC, which allowed a mobile node to transmit in burst once it gained access to the wireless medium. The protocol utilized a Least Mean Square (LMS) filter that continuously evaluated the RSSI values of received ACKs from its temporary parent. If any persisting deterioration in the link quality were detected, a handover procedure was initiated. Similarly to MA-MAC, if the number of neighboring static nodes was small, then the probability of discovering a new static relay node reduced significantly.

Kuntz et al. [102] proposed X-Machiavel, a X-MAC-based solution. X-Machiavel prioritized data packets from mobile nodes over those of static devices. It allowed mobile sensors to “steal" the wireless medium from a static node that had gained it earlier. When a mobile node expected to transmit a data packet, it first sampled the medium. If no signal were detected, it followed the standard procedure (i.e., preamble period prior the data packet). Else (i.e., detection of another preamble), it waited for the end of the ongoing preamble being sent by a static node in order to further use the reserved medium before the transmitting device. Prioritizing mobile nodes over static ones eventually led to increase both 1-hop and end-to-end delays and X-Machiavel might thus considered as a “non fair” contention based protocol.

Similarly to X-Machiavel, mobile nodes using MoX-MAC [103] sampled the medium hoping to detect any ACK packet sent to a static node. Any detection led the node to wait until the end of the scheduled transmission. Afterwards, it transmitted its data packet to the transmitting static node. Otherwise, if no ACK packet were detected, the default X-MAC procedure was followed. After the transmission of its data packet, the emitting static node could keep its radio turned on for potential transmission from a mobile node. The efficiency of this approach strongly depended on the communication frequency between the static nodes. Moreover, if no data packet came from any mobile node, static nodes unnecessarily consumed energy by keeping their radio on.

MOBINET [104] allowed mobile nodes to overhear transmitted packets in order to detect potential surrounding static nodes. Upon data transmission, mobile nodes sent a data packet in unicast to one of the destination addresses listed in the so-built neighborhood table. MOBINET came with both a random and a selective method. In the first approach, the next-hop node was randomly selected among the ones available in the neighborhood. The second aimed at using the "best" neighbor (e.g., number of hops, depending on the routing protocol).

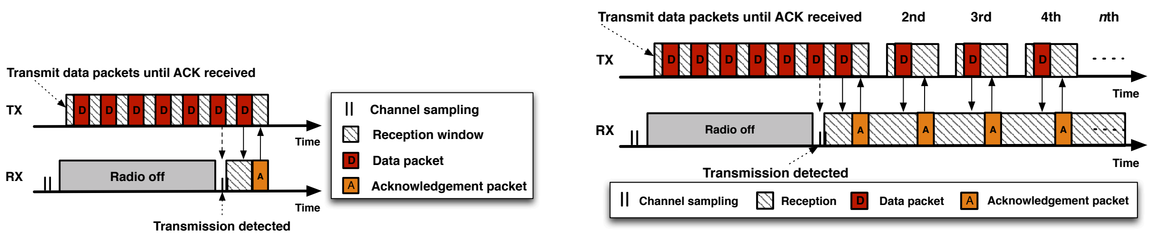
MAC protocol	**Advantages**	**Drawbacks**
MA-MAC [97], MARI-MAC [99], MX-MAC [91]	• traffic independent • handover mechanism integrated	• reactive protocols • inaccurate proximity estimation • network density dependency • designed for very small networks
X-Machiavel [102]	• traffic independent • hybrid protocol • overhead minimization (preamble-less)	• underlying protocol dependency • proportion of mobile to static nodes dependency • non-fair contention-based protocol
MoX-MAC [103]	• proactive protocol • overhead minimization (preamble-less)	• traffic dependent (passive protocol) • unnecessarily consume energy (for static nodes)
MOBINET [104]	• proactive protocol • optimal next-hop selection	• traffic dependent (passive protocol) • increase of idle listening (energy consumption)
M-ContikiMAC, ME-ContikiMAC	• proactive protocol • traffic independent • independency from underlying protocol • overhead minimization	• inefficient under intermittent link connections • inefficient handover & recovery procedures

Table 4.1: Summary of state-of-the-art mobility-aware LPL contributions.

Most of the previously presented protocols were not satisfying our objectives of addressing bursty traffic in mobile environments, in a highly proactive manner and by attaining low 1-hop and end-to-end delay values under mobile environments, as summarized in Table 4.1. We thus proposed several approaches to allow for low-power and low-delay neighbor discovery while investigating seamless handovers in the network.

4.2 Extending ContikiMAC for mobility-aware WSNs

Originally, ContikiMAC [60] provided two types of transmissions, the so-called unicast and broadcast. A unicast operation consists in a transmitter repeatedly sending its data packet until receiving a link layer acknowledgment once the intended receiver wakes up (Figure 4.1(a)). Broadcasting nodes were not expecting any acknowledgment and thus sent their packets during the entire sleeping period.



(a) ContikiMAC in unicast mode.

(b) ContikiMAC in burst mode.

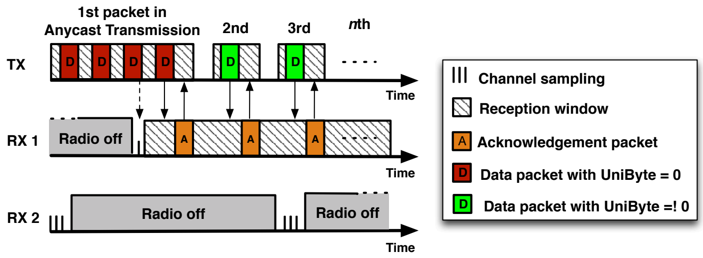
Figure 4.1: Operations of ContikiMAC.

As depicted on Figure 4.1(b), ContikiMAC also allowed receivers to stay on if senders have indicated more data packets to come. Upon reception of a so-flagged data packet, a receiver switched into Carrier Sense Multiple Access (CSMA) mode until it received the last data of the burst (i.e., non-flagged packet). After each data transmission, the sending node was waiting for a ACK before emitting the next data packet [94].

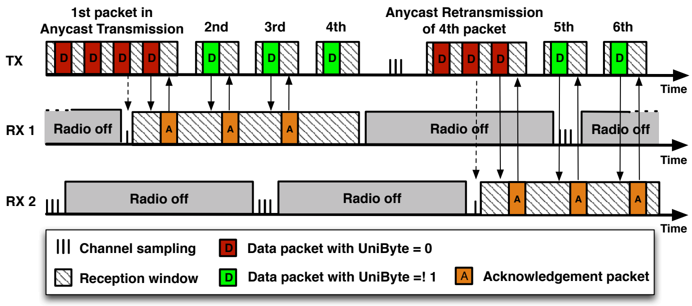
4.2.1 M-ContikiMAC: Introducing anycast data packets from mobile nodes

Assumed as unaware of the underlying routing topology, burst-transmitting mobiles nodes had to either use (costly) broadcasted packets or first transmit in broadcast in order to discover a parent before switching to further unicast transmissions. We thus introduced anycast transmissions, in order to reach all potential receivers with a same destination address [15]. Packets were then unicasted to the first potential forwarder acknowledging the corresponding packet, as illustrated in Figure 4.2(a).

Furthermore, we considered bursty transmissions, thus inducing packets with an activated burst notification flag. The first packet includes an additional flag, from now UniByte, which indicates that transmitter is searching for a temporary next-hop when is equal to zero. Upon first reception of an acknowledgment, the transmitter identifies the corresponding receiver as its temporary next-hop, thus setting UniByte to one and transmitting its remaining packets to this node. For the sake of stability, only static nodes were allowed to respond to anycast transmissions from mobile nodes. Finally, the new temporary next-hop shall keep the radio on in order to receive the further packets, as originally designed by ContikiMAC.



(a) Basic mode: TX sends first in anycast, then in unicast.



(b) Connection recovery.

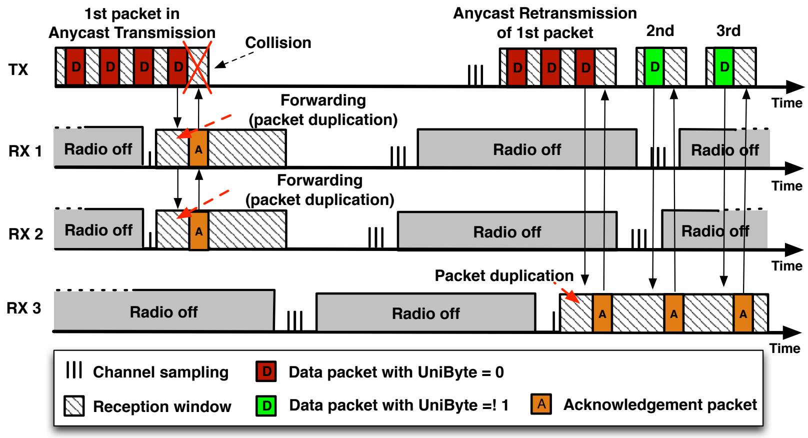
Figure 4.2: Basic operations of M-ContikiMAC.

4.2.2 ME-ContikiMAC: Handling link disconnections and duplicate messages

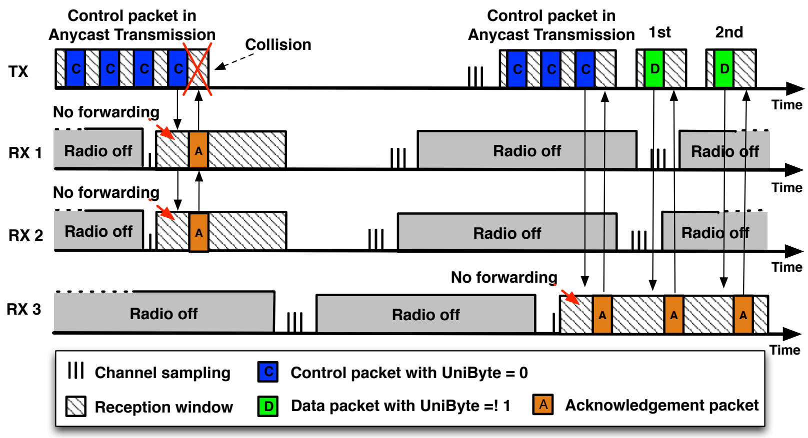
Due to node mobility or radio conditions, link disconnection may occur. This loss of next-hop is detected by a mobile node as soon as one acknowledgment is missing, thus triggering the transmission of the last non-acknowledged packets back in anycast mode (i.e., for discovery of a new next-hop). As shown in Figure 4.2(b), once TX node is out of RX1's range (non reception of its fourth packet), it switches back to anycast transmissions and manages to wake-up a new surrounding node, RX2.

Moreover, an initial anycast transmission may lead to multiple simultaneous acknowledgements from distinct neighbors that would result in collisions at the sending node. While the first data packet would be forwarded by those receivers, the sender would postpone the burst transmission for the following preamble cycle, thus causing duplicate messages throughout the network (Figure 4.3(a)). Hence, the traffic in the network, congestion, channel occupancy and the competition of the medium access all increase, which in turn enlarges the probability of packet retransmissions due to the potential collisions. In the meantime, the receivers (RX1 and RX2) are unaware of their colliding acknowledgments and keep their radio on, waiting for more packets to come.

In order to mitigate such duplications and avoid energy wastage at receivers, we allowed senders to transmit one additional packet upfront [14]. This control packet embeds a UniByte equals to zero and has a time-to-live of 1, thus guaranteeing no further forwarding. The burst transmission will start upon reception of an acknowledgment, whose sender will be identified as the temporary next-hop by the sender, as depicted in Figure 4.3(b).



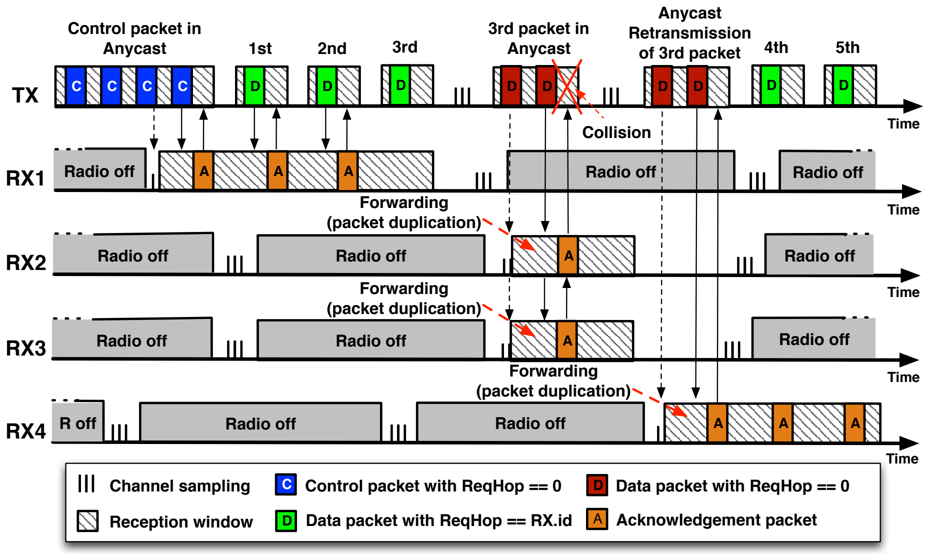
(a) When two or more nodes have the same wake-up slot, then a packet may be received by two or more nodes, that will in turn forward it. This situation leads to packet duplication at the sink.



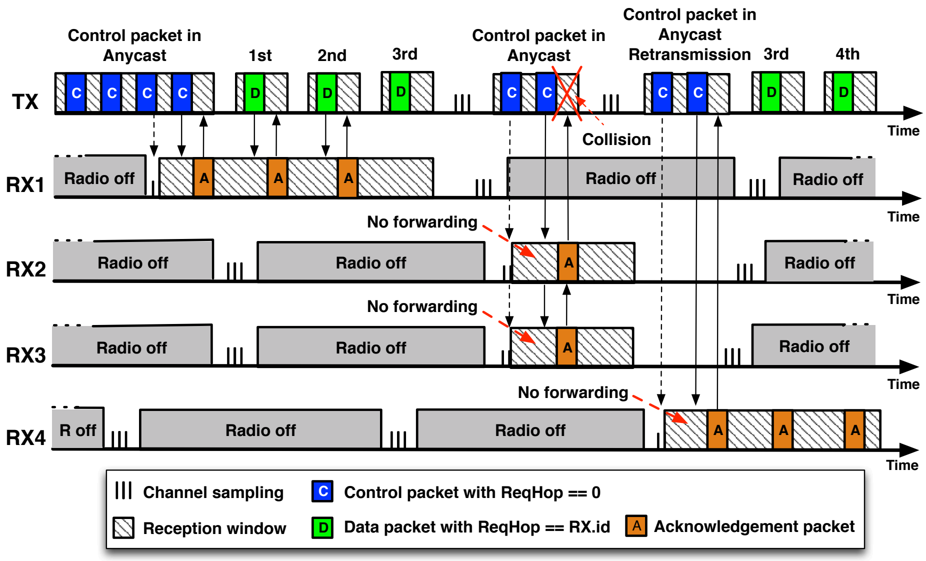
(b) ContikiMAC in enhanced mode: By introducing control packets before the burst transmissions we manage to reduce the duplications at the sink.

Figure 4.3: Packet duplication phenomenon (left) and proposed ME-ContikiMAC (right).

Colliding acknowledgments of the control packet simply result in the sender postponing its neighbor discovery. Note that in case of link disconnection during the burst, the last non-acknowledged data packet should be repeated only once a new control packet has been successfully sent (i.e., new next-hop discovered), as shown in Figures 4.4.



(a) To recover a connection by transmitting a data packet, we may end up with duplications if two or more nodes will sample the medium simultaneously.



(b) ME-ContikiMAC in enhanced connection recovering depiction: Upon a link disconnection the mobile node transmits a control packet upfront to avoid duplications.

Figure 4.4: Scenarios where packet duplication issues arise (left) and ME-ContikiMAC with packet duplication control mechanisms (right).

4.2.3 MobiDisc: Enhancing neighbor discovery and improving handovers

We investigated seamless handovers in order not to interrupt a (burst) transmission upon next-hop changes that inevitably occur due to node movement, link fluctuations and disconnections. We assumed that static neighbors could provide valuable information (e.g., battery power, number of hops to the sink) that would help surrounding mobile nodes select the best next-hop without knowing anything about the routing infrastructure. Furthermore, once a given next-hop has been selected, more interesting nodes may show up in the vicinity of the mobile nodes. We thus proposed that mobile nodes keep discovering other potential forwarders, based on regular updates from surrounding nodes. We introduced MobiDisc for mobile nodes to discover either the whole neighborhood or only a subset of nodes located in the vicinity, while enabling seamless handovers in the network [16, 17].

Standard neighborhood discovery of MobiDisc

MobiDisc enables information exchange during the neighborhood discovery phase of mobile nodes. This information exchange may adapt to the application layer requirements. As for other propositions of this manuscript, MobiDisc is compliant with any preamble-sampling protocol. In our case, we designed it in conjunction with ContikiMAC, which, as earlier described, comes with a burst handling mechanism to anticipate high traffic loads in the network.

Let us assume a mobile node that expects to transmit n-packets in row. Due to its mobility nature, the node is not aware of the static nodes that are (or will be) located in its transmission range, and even more about their distance (i.e., in terms of hops) to the sink. First the mobile node performs a neighborhood discovery with anycasted control packets, before transmitting its n packets in a bursty manner.

Figure 4.5 shows that, unlike M-ContikiMAC, MobiDisc sends control packets during a whole sampling period thus aiming at a complete neighbor discovery. The mobile node can then select the best next-hop according to the available metric (e.g., remaining battery power, number of hops to the sink station, value of some link quality indicators). Note that after acknowledging the control packet, static nodes turn off their radio, for energy saving purposes. For the mobile node, the next sampling period will be devoted to the burst transmission of the n packets to the previously selected static node.

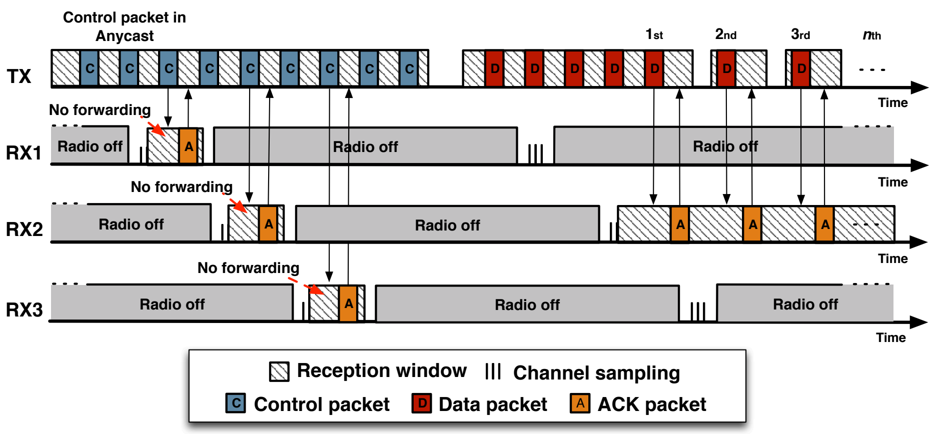


Figure 4.5: MobiDisc (default mode): after a neighborhood discovery with anycasted control packets, TX chooses RX2 and starts sending data packets right after its next sampling period.

First ACK Next-hop (FAN)

We then worked on a faster neighbor discovery that would allow mobile nodes to opt for the first next-hop to have acknowledged a control packet (named First ACK Next-hop, FAN mode), while keeping updated about other surrounding and potentially better neighbors. After this initial period (similar to M-ContikiMAC), a mobile node initiates the transmission of n packets and also listens to further notifications of other surrounding nodes. Later, the mobile node may assess a new best candidate and potentially perform a handover.

Indeed, as observed on Figure 4.6, the mobile node moves from a neighbor 3-hop away from the sink (i.e., RX1) to another one only 1-hop away from the sink (i.e., RX2). If TX will rate the RX2 as the optimal next-hop among RX1, RX2 and RX3, it will then perform the handover by switching from RX1 to RX2, and thus, TX will transmit the following data packet to RX2.

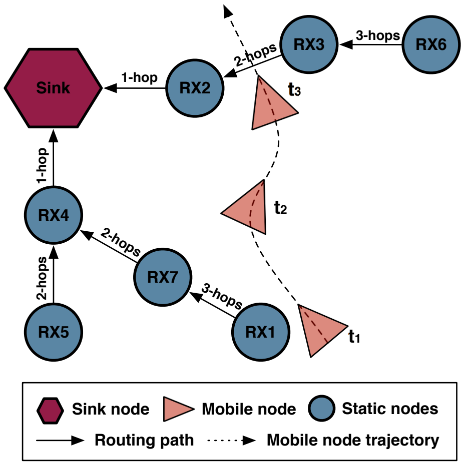


Figure 4.6: FAN mode: Upon reception of notification packets, a mobile may change next-hop.

It is important to mention that in this case RX1 (and RX3) after expiration of default timeout (i.e., between 20 to 30 ms) will turn its radio off to save energy, like with any other preamble-sampling MAC protocol. The functionality of our FAN mode is depicted in Figure 4.7.

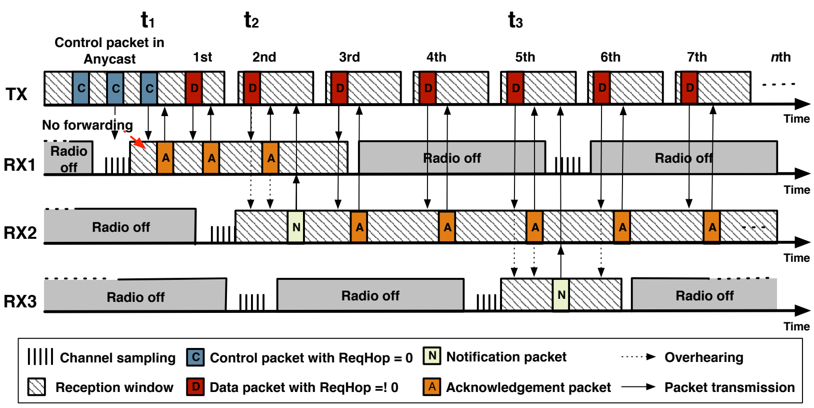


Figure 4.7: FAN mode: static nodes here announce a min-hop metric value to mobile node TX, thus allowing it to switch its next-hop (topology being as on Figure 4.6).

Fast recovering mechanism (FRM)

The FAN optimization allowed for quicker connections of mobile nodes to the static infrastructure, yet imposing frequent handovers for further changes of next-hop. This led to non negligible energy costs and higher rates of packet losses. We addressed these issues by introducing a fast recovering mechanism (FRM). As illustrated on Figure 4.8, a mobile node (TX) should detect the disconnection with RX1 as soon as possible, while starting to look for a new next-hop (RX2).

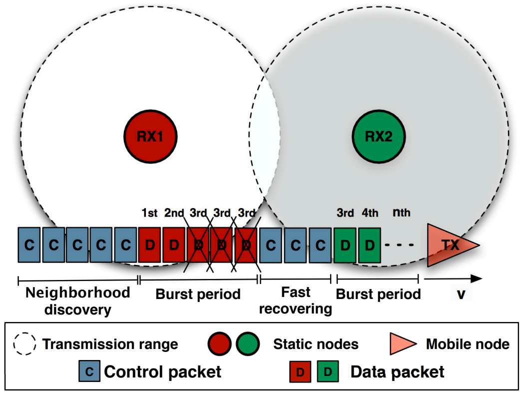


Figure 4.8: A recovery use case integrated in MobiDisc (default mode).

As can be observed from Figure 4.9, the third data packet from mobile node TX is never acknowledged by its temporary receiver RX1. After the default three tentatives, TX cancels its transmissions, and postpones them for the following preamble round. Hence, it will waste time for at least one complete sampling period (e.g., 125 ms) before initiating a new neighbor discovery process. As a result, MobiDisc could induce high handover delays, especially if the nodes were configured with long preamble-sampling frequencies such as 500 ms.

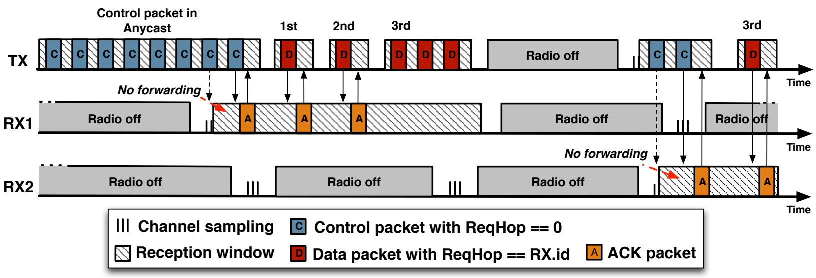


Figure 4.9: Handover delay due to the default setup of MobiDisc: upon link disconnection with RX1, mobile node (TX) postpones its transmission for the following preamble round.

In order to improve the previously discussed handover and reconnection delay issues, we designed a Fast Recovering Mechanism (FRM), integrated in MobiDisc. By employing the FRM, mobile nodes have priority to the medium access over static ones. In particular, once a mobile node detects the network disconnection (after transmitting repeatedly the data packet for a predefined time), it immediately performs a next-hop discovery (by repeatedly transmitting control packets) during the same preamble cycle. By this way, we aim at reducing the 1-hop delay from mobile to static node (see Figure 4.10).

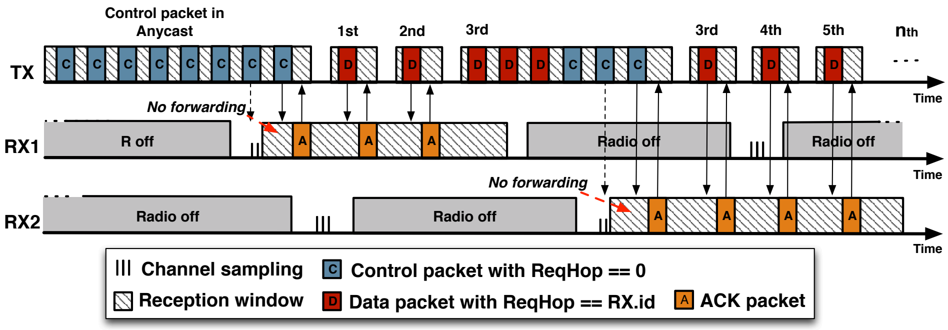


Figure 4.10: Enhanced version of the handover mechanism of MobiDisc: mobile node (TX) continues transmitting control packets without turning its radio OFF.

4.3 Main results

The following results were obtained from a topology of 40 fixed nodes (sink included), either uniformly (i.e., grid) or randomly distributed over a 50 $\times$ 40 m area, with a network degree of 13.6 in average (similarly to dense wireless lighting control networks [105]). Static nodes were transmitting 1 pkt per 30 sec, based on a Constant Bit Rate (CBR). Mobile nodes were configured to transmit bursts of 16 packets every 90 sec, thus leading to more than 8000 $pkts$ transmissions in total, each including 33 $bytes$ that corresponds to all necessary information for MAC, routing and application operations (e.g., node ID, packet sequence, burst and ReqHop flags, sensed values). For all nodes, the MAC sampling frequency was set to 125 ms and a maximum of three retransmissions was defined. We used an omnidirectional antenna with a transmission power of -10 dBm, thus imposing multihop communications between the mobile nodes and the sink (up to five hops). At the routing layer, we relied on a sink-routed gradient protocol (i.e., with as the number of hops towards to the sink as metric). BonnMotion [106] allowed us to use a random waypoint mobility model in order to move 8 mobile nodes. We used both low and high speed (i.e., from 0.5 m/s to 2 m/s and 2 m/s to 8 m/s), thus representing respectively a human walk and a typical jogging speed. Each simulation lasted 54 min.

M-ContikiMAC and ME-ContikiMAC were implemented on top of Contiki OS, by using COOJA [89] with Sky motes. We compared our proposals with state-of-the-art protocols such as MoX-MAC and MOBINET, using both selective (MOBINET-S) and random (MOBINET-R) methods for next-hop identification. Note that none of them does allow for direct synchronization after successful transmissions (i.e., avoiding preambles). In order to provide fair analysis and thorough comparative study, we thus deactivated the phase-lock optimization function from the default configuration of ContikiMAC.

Moreover, in order to emphasize the importance of seamless handovers, the results related to MobiDisc were obtained from more traffic-demanding mobile nodes (32 packets every 120 sec) with smaller transmission ranges (-12 dBm). Finally, for readability purposes, MobiDisc is here compared to the most relevant protocols only (i.e., MoX-MAC and ME-ContikiMAC), as identified per other provided results.

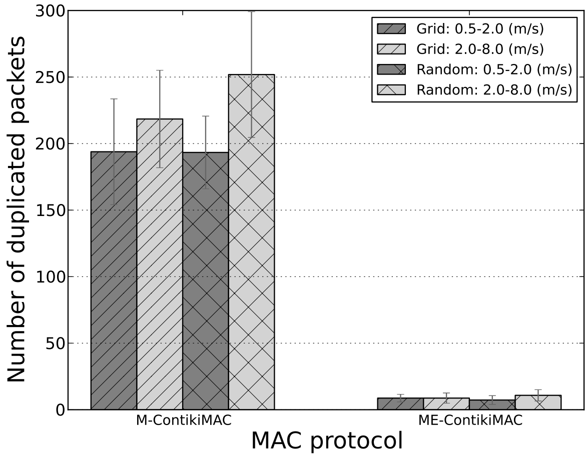
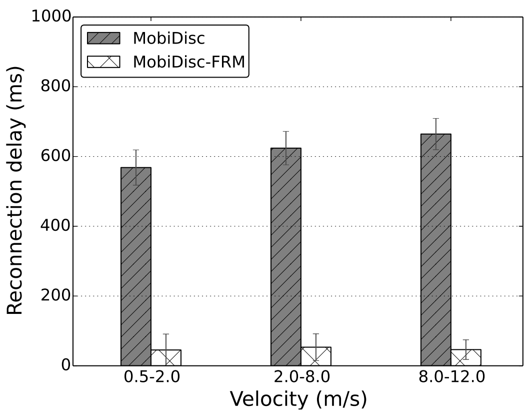


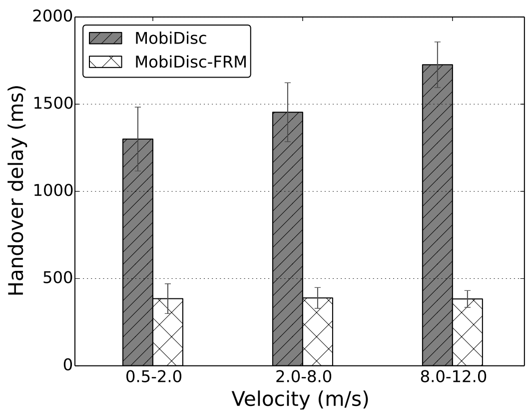
Figure 4.11: Total number of packet duplications at the sink.

4.3.1 Mitigating packet duplications

As observed on Figure 4.11, uncontrolled anycast transmissions may cause multiple packet receptions in the network. After introducing such packets with M-ContikiMAC, ME-ContikiMAC allowed to decrease packet duplications, thus limiting network traffic congestion, channel occupancy and medium access competition increase. Indeed, duplications were reduced by more than 90% comparing to M-ContikiMAC, thus avoiding a large amount of unnecessary packets and potential collisions.



(a) Average reconnection delay, from mobile to any static node.



(b) Average handover delay, from mobile to any static node.

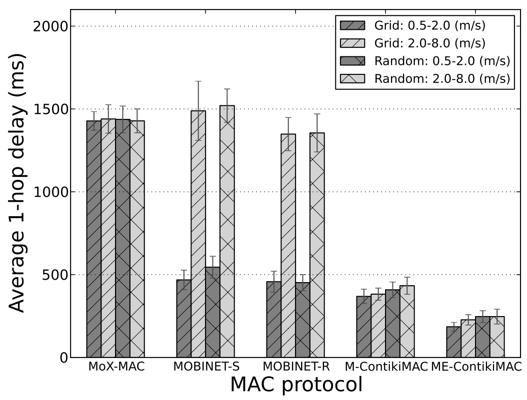
Figure 4.12: Reconnection and handover delays of MobiDisc in grid scenario (default mode, with and without integration of fast recovering mechanism).

4.3.2 Reducing handover delays

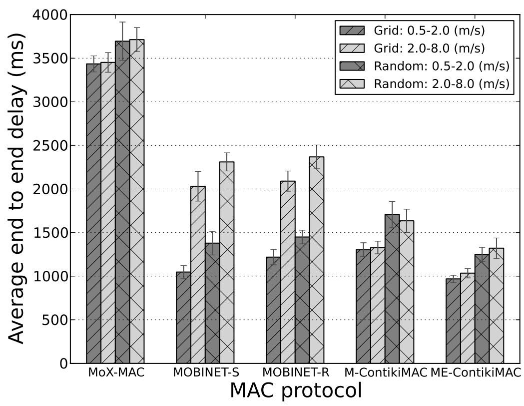
The average reconnection time (i.e., required to establish a new link since the disconnection) and handover delays were evaluated, for each packet transmission from any mobile to any static node. As detailed in Section 4.2.3, in our MobiDisc proposition, we allowed mobile nodes to be aggressive by directly transmitting control packets upon non acknowledgment of a data packet (i.e., during the same preamble cycle). Figures 4.12(a) and 4.12(b) confirm the efficiency of our fast recovering mechanism (FRM), since both reconnection and handover delays could be considerably reduced for all velocities over a grid topology.

4.3.3 Reducing 1-hop and end-to-end delays

Figures 4.13(a) and 4.13(b) illustrate the average 1-hop (from mobile to any static node) and end-to-end (from mobile to sink node) delay per packet transmission. The presented 1-hop and end-to-end delay include the channel sampling period, initial back-off, potential congestion back-off, potential retransmission delay and the transmission time of the preamble. All Figures confirm that high velocity scenarios lead to increased difficulties for link establishment between the mobile and static node. Logically, higher speed induces more frequent connections and disconnections. We also observe that, considering end-to-end delay, all protocols perform worse in random topologies. This phenomenon takes place due to the potential bottleneck links that are more prone to appear in random topologies, having as a result nodes to handle heavy traffic.



(a) Average one hop delay (mobile to static).



(b) Average end to end delay (mobile to sink).

Figure 4.13: Delay performance.

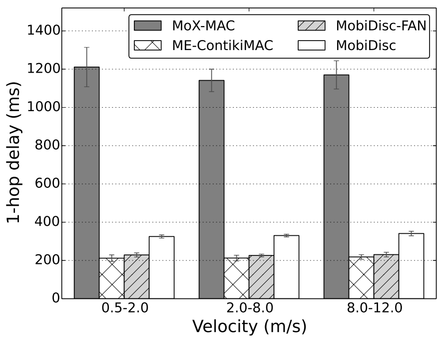
The poor performances of MoX-MAC can be explained by the fact a mobile node first overhears the whole transmission between the static nodes, and later it transmits its data packet to the detected sender. Similarly, both methods of MOBINET also suffered from high velocities, most probably since a mobile nodes fail to overhear the transmissions from neighboring devices. On the other side, MOBINET showed promising results in end-to-end delay for low speed scenarios, and more specifically, for the selective version of MOBINET, since it was selecting the best parent among the list of the potential next hop nodes (see Table 4.2). As a result, it picked up the node closest to the sink in terms of hop.

Scenario	MoX-MAC	MOBINET-S	MOBINET-R	M-ContikiMAC	ME-ContikiMAC
Grid: $0.5 - 2.0$ (m/s)	3.34 (0.04)	2.53 (0.05)	2.99 (0.04)	3.46 (0.06)	3.26 (0.07)
Grid: $2.0 - 8.0$ (m/s)	3.31 (0.05)	2.47 (0.05)	2.95 (0.06)	3.48 (0.09)	3.28 (0.04)
Random: $0.5 - 2.0$ (m/s)	3.64 (0.09)	3.09 (0.13)	3.45 (0.09)	4.02 (0.29)	3.70 (0.22)
Random: $2.0 - 8.0$ (m/s)	3.65 (0.09)	2.95 (0.15)	3.45 (0.09)	3.97 (0.22)	3.71 (0.10)

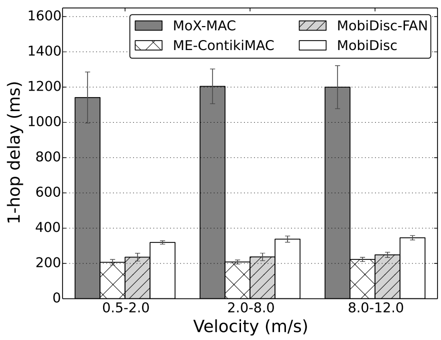
Table 4.2: Average (along with confidence interval) number of hops, from mobile to sink.

Compared to other protocols, ME-ContikiMAC significantly improved both 1-hop and end-to-end delays for all considered scenarios. Especially, it reduced up to 60% the performance in high speed scenarios, mainly because of the prioritization of mobile nodes. We also reduced unnecessary transmissions in the network, which decreased potential collisions, and consequently retransmissions that had a major impact on the delay performance.

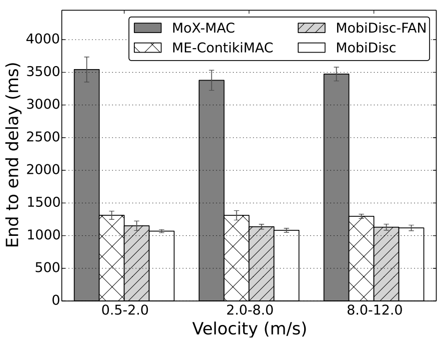
Furthermore, as can be observed from the Table 4.2, all solutions had more or less the same amount of hops (the lowest being achieved by MOBINET, because of its next-hop handling at MAC layer), thus meaning that the end-to-end delay reduction is independent to this metric.



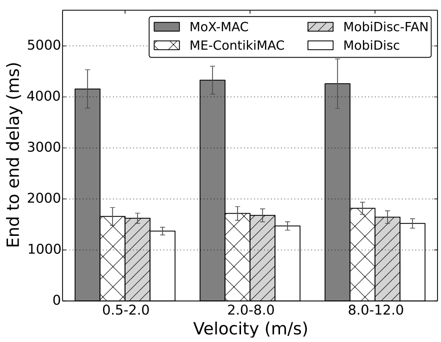
(a) Average 1-hop delay, from mobile to any static node, grid topology.



(b) Average 1-hop delay, from mobile to any static node, random topology.



(c) Average end-to-end delay, from mobile to sink station, grid topology.



(d) Average end-to-end delay, from mobile to sink station, random topology.

Figure 4.14: 1-hop and end-to-end delays of MobiDisc (default and FAN modes), when compared against MoX-MAC and ME-ContikiMAC.

Based on these results, we compared both MoX-MAC and ME-ContikiMAC to our MobiDisc solution (under both default and FAN modes, FRM being activated). Figures 4.14(a), 4.14(b), 4.14(c) and 4.14(d) illustrate the average 1-hop (from any mobile to any static node) and end-to-end (from any mobile to sink node) delay per data packet transmission.

As expected, MobiDisc-FAN and ME-ContikiMAC attained the best performance regarding 1-hop delay, since these schemes opportunistically initiateed the neighborhood discovery, to the first static node that acknowledged the corresponding anycast control packet. Those results confirmed the bad performances achieved by MoX-MAC, especially with smaller network density and increased traffic from mobile nodes. Concerning MobiDisc in default mode, mobile nodes repeatedly transmitted control packets during the whole preamble period (e.g., 125 ms), thus explaining the increased 1-hop delay, when compared to MobiDisc in FAN mode.

However, when considering end-to-end delay (i.e., from any mobile node to the sink station), the default operation of MobiDisc outperformed all other propositions, for all considered scenarios (i.e., velocities and topologies), thanks to its efficient next hop selection (i.e., choosing a static node closer to the sink station).

4.3.4 Limiting energy consumption

Figure 4.15 shows the average energy consumption per second for the whole network, with for both grid and random topologies. We could observe that the overhearing procedure had a straightforward impact on energy dissipation, as ME-ContikiMAC consumed less energy (i.e., 1 mW in average) network-wide when compared to MoX-MAC, and both selective and random-based MOBINET protocols.

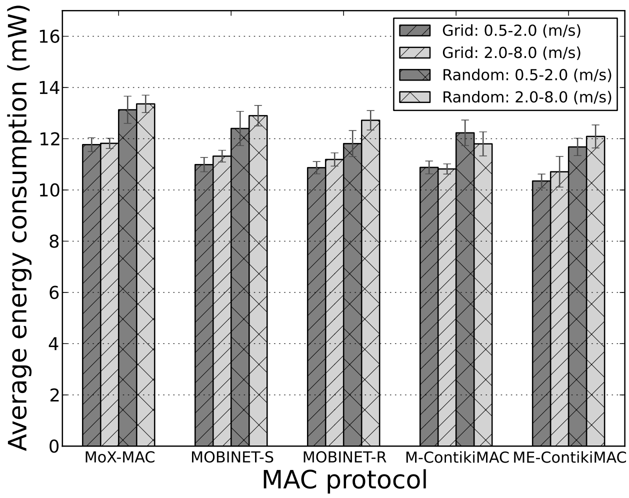
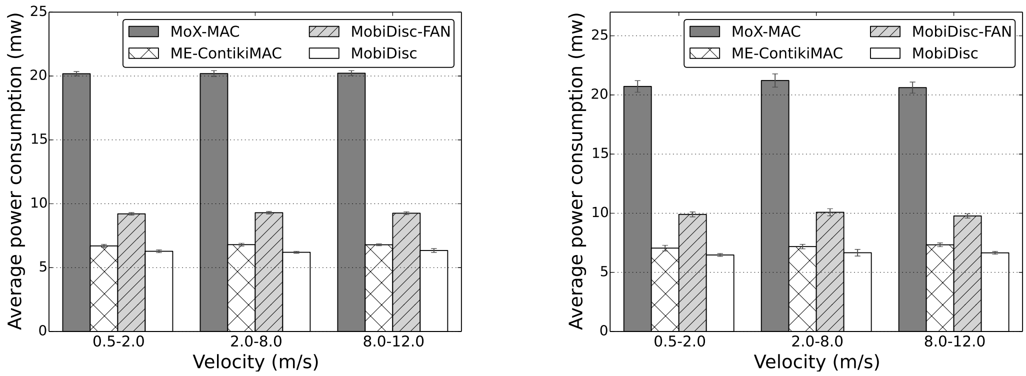


Figure 4.15: Average energy consumption (both in regular grid and random topologies).

In all cases, MobiDisc simulations exhibited a lower energy consumption, as shown on Figures 4.16(a) and 4.16(b). Indeed, MobiDisc (default mode) reduced the average number of hops toward the sink station , and thus, the total packet transmissions in the network. However, even though achieving quick 1-hop connections, MobiDisc-FAN comes with high energy consumption due to its multiple CCA checks (i.e., five) and non selective next-hop determination. Finally, we should take into account the fact that schemes such as MoX-MAC and MobiDisc-FAN are based on overhearing technique, which actually means the radio remains active for more time.



(a) Grid topology.

(b) Random topology.

Figure 4.16: Energy consumption of MobiDisc (default and FAN modes), when compared against MoX-MAC and ME-ContikiMAC.

4.4 Conclusion and future works

In this Chapter, we presented several of our contributions to handle mobile nodes within IoT networks that rely on preamble-sampling. After a detailed analysis of ContikiMAC, we decided to focus on burst transmissions that necessarily come out of mobile nodes. The connectivity of these devices to the infrastructure of static nodes imposed to take maximum advantage of 1-hop connections when they could be established. We thus proposed M-ContikiMAC for a proper neighbor discovery that did not require any broadcast of association requests. We then mitigated the multiple duplicate messages by introducing ME-ContikiMAC and demonstrated that two different configurations (i.e., statical oriented and mobile oriented) of the same MAC protocol could be combined, so that mobile nodes could smoothly communicate within a static network, without deteriorating the quality of communications between fixed devices. During our research on that topic, we did our best to remain compliant with existing proposals. More specifically, all described mechanisms can be applied to most of preamble-sampling protocols (e.g., X-MAC) and were implemented on top of ContikiMAC. When compared to M-ContikiMAC and other state-of-the-art protocols (i.e., MoX-MAC, MOBINET), ME-ContikiMAC enhanced the overall network performance significantly.

We then focused on this contribution and investigated seamless handovers as mobile nodes inevitably spend most of their time to move across the static nodes. We introduced MobiDisc and its two different neighbor discovery methods, the first one emphasizing the need for shortest path to the sink without a priori knowledge of the underlying routing topology (default mode) and the second one allowing for faster next-hop selection (FAN). A fast recovering mechanism also ensures reactivity upon link failure. This complete solution provided us with promising results in terms of both reconnection and handover delays (i.e., more that 55% reduction when compared to ME-ContikiMAC) while allowed for uninterrupted sensing.

These contributions were all proposed during Georgios Z. Papadopoulos's PhD and were published in several conferences and journals [14–17].

Considering these investigations, several research leads would be interesting to follow. The first one is related to engineering aspects, as mobile robots enter IoT testbeds and allow for enhanced experiments around mobility. Our quite low ratio of mobile nodes (i.e., 8 over 48 node topologies) could be reproduced in such facilities. Reproducible trajectories and non random movements (e.g., mobile devices acting as collecting sinks) would allow for further proposals around energy-efficient and robust communications for mobile IoT. As some recent contributions assume paths of moving sensors to be known a priori [107], enhanced anticipation of handovers are becoming realistic, especially as data traffic can also be anticipated in some specific IoT scenarios (e.g., Industrial IoT networks). Furthermore, we assumed that some information of the routing layer could be made available at the MAC layer (e.g., number of hops to the sink). Conversely, examining the performances achieved by such routing schemes relying on our enhanced MAC layer also seems promising.

Chapter 5

Conclusion and perspectives

Contents

5.1 Conclusion

As synchronized protocols seemed adapted to specific scenarios (e.g., a priori know traffic patterns, static topology), we decided to target low-power listening (LPL) mechanisms under varying topology and traffic assumptions, having in mind some specific applications and deployments (e.g., remote healthcare, wildlife monitoring). In this context, several LPL based MAC protocols were well adapted to pre-configure the MAC layer of the sensor nodes prior to a deployment (e.g., B-MAC, X-MAC) but their operation in a dynamic scenario (e.g., traffic, mobility) would have required periodic re-computation of the adjustable parameters while the network was running.

Since 2008, I have been lucky to work jointly with several people on this topic. Especially, the PhD periods of Romain Kuntz, Julien Beaudaux and Georgios Z. Papadopoulos have led to the proposition of multiple consistent and performing solutions, while adopting a theory to practice approach.

As reviewed in Chapter 2, our works around heterogeneous LPL configurations in connected IoT networks highlighted that involved nodes could opt for various sampling periods, without endangering multihop communications. In this context, the PhD thesis of Julien Beaudaux detailed how to take advantage of various information available at each node in order to provide localized solutions with limited control overhead. Sensing-only nodes were introduced and allowed to adopt specific LPL configurations for large subsets of devices, while guaranteeing network connectivity and preventing from any isolated node. We investigated and combined several techniques and algorithms (e.g., LMST construction, use of routing tree, application criterions for heterogenous sleep depth) in order to ensure efficient data collection while saving energy. Our solutions provided a better compromise between energy-efficiency and networking performances than any homogeneous LPL configuration.

We then considered the dynamics that WSN would undoubtedly have to face (Chapter 3). Two of our main contributions regarding the auto-adaptation of preamble-sampling MAC protocols (i.e., BOX-MAC and T-AAD) relied on agreements to enable energy-efficient links (i.e., short sampling periods and preambles used on both ends). Upon considering traffic anticipation, we also proposed a stable, lightweight and traffic independent solution and further examined to what extent such auto-adaptive mechanisms could help mitigate packet duplication induced by the routing layer. We later introduced deafness that allowed nodes to adopt long sampling periods, thus reducing the probability of catching preambles of unnecessary transmissions. Such methods could benefit from previous overhearing of communications in the neighborhood in order to define the local number of potential data relays. Overall, we showed that local configurations made on available information from control-packet transmissions (e.g., routing construction) could leverage the unnecessary overhead, without endangering packet reception rates.

Eventually, Chapter 4 was devoted to several of our contributions to handle mobile nodes within IoT networks. We focused on burst transmissions that necessarily come out of mobile nodes. We proposed a request-free neighbor discovery, while mitigating duplicate messages by prioritizing mobile nodes and without deteriorating the quality of communications between fixed devices. Our proposals remained compliant with most LPL protocols and exhibited reduced packet duplications, channel occupancy and delays. As mobile nodes inevitably spend most of their time to move across the static nodes, we then studied seamless handovers and proposed two different neighbor discovery methods that either allowed for quick connection to the static infrastructure or reduced delays towards the sink station.

5.2 Perspectives

This manuscript has presented most of my research activities over the last decade. Of course, there are still many open issues to be investigated around LPL schemes (e.g., efficient multi-channel LPL [108], RPL routing over low-power listening devices [109]). However, recent advances at the physical layer have questioned the relevance of preamble-sampling solutions. Available Wake-up radio (WuR) chips can now be added to existing IoT devices in order to allow for on-demand wake-up of both microcontroller and wireless units. In this new technological context, few studies provide information related to the different performance provided by WuR when compared to traditional MAC approaches. Obviously the wake-up call imposed by WuR comes at the cost of a delay, but it is expected to remain smaller than the preambles required by preamble-sampling approaches. As detailed in [110], even though some questions are still unanswered (e.g., behavior of WuR in noisy environments, achievable performances in dynamic scenarios), the use of WuR should provide significant improvements for networks that rely on LPL-based MAC strategies. Moreover, most of today's deployments of IoT networks are requiring guarantees and bounded delays and consumption (e.g., industrial applications), which are still difficult to obtain from preamble-sampling solutions.

Alongside our contributions to low-power listening mechanisms, we have conducted several studies in related fields of the IoT domain. Those other works have become research items I would like to further investigate. They range from performance evaluation to fault-tolerance, while including Industrial Internet of things (IIoT).

5.2.1 Performance evaluation

As simulations are considered as not sufficient to demonstrate the consistency of a solution (e.g., simplified modeling for tractability purposes), the availability of low-cost devices have led researchers and engineers to enrich performance evaluation with testbeds. In [18, 19], we compiled a large set of statistics from 674 articles published in the top representative ad hoc and wireless sensor networks related conferences (i.e., ACM/ IEEE IPSN, ACM MobiCom, ACM MobiHoc, and ACM SenSys) over the 2008–2013 period. Our goal was to explore the role of simulators and testbeds from the theoretical analysis of a model and throughout the protocol development procedure. Indeed, while simulators allow for an easier, faster, and less expensive process of validation, researchers have overcome the technical challenges and economical barriers of real-world deployment to perform a thorough experimental evaluation of their ideas in wide-scale platforms (e.g., custom and small facilities thanks to low cost hardware, open and large-scale testbeds). Simulation evaluations are assumed to allow for reproducible setups, thus producing scientific results that can be reproduced and verified. As we discovered that more and more researchers were relying on custom or open testbeds for their performance evaluation campaigns, we emphasized that the presented results had a low reproducibility level (i.e., 16.5%). We thus raised the question of whether simulations and experiments were leading to scientific results (i.e., reproducible from repeatable setups) or proofs of concepts only.

Regarding experiments, even though the complete setup is provided, it is often difficult or simply impossible to reproduce the conditions of the original experience (e.g., radio environment impacted by other surrounding and uncontrollable networks or perturbation sources). We greatly benefited from our partnership within the FIT Equipex consortium, which allowed the University of Strasbourg to host a large-scale open IoT testbed [8]. Our initial works on the so-built platform helped us apprehend the complexity of such evaluation methods [20]. When studying the X-MAC protocol, we had tried to better understand and characterize those radio environments in order to define stable radio topology over which experiments of a MAC layer would make sense (i.e., observed variations depending on MAC parameters only) [21]. In [22, 23], we thus further explored the role of simulators and testbeds in the development procedure of protocols or applications for Wireless Sensor Networks (WSNs) and Internet of Things (IoT). We investigated the complementarity between simulation and experimentation studies by evaluating latest features available among open testbeds (e.g., energy monitoring, mobility). For instance, mobility is considered as a service that testbeds should provide [111] but requires complex implementation to make it repeatable (i.e., trajectory planning, indoor localization, preserved radio environment).

We showed that monitoring tools and control channels of testbeds could allow for identification of crucial issues (e.g., energy consumption, link quality) and we identified some opportunities to leverage those real-life obstacles. Our objective was to enable multiple instances of a same experimental setup over stable and finely controlled components of hardware and real-world environment in order to obtain reproducible results. Future works consist in guaranteeing stability of hardware and environment components over time, thus, turning the unexpected failures and changing parameters into core experimental parameters and valuable inputs for enhanced performance evaluation. Similarly to Guix-HPC[8], an effort to optimize GNU Guix for reproducible scientific workflows in high-performance computing (HPC), we aim at designing building blocks that could be configured and used within IoT testbeds (e.g., sets of topologies, mobility and traffic patterns, interference maps) in order to conduct repeatable setups and publish reproducible results.

[8] `https://guix-hpc.bordeaux.inria.fr/`

5.2.2 Industrial Internet of Things

Alongside the numerous contributions around asynchronous MAC protocols, the IEEE 802.15.4 standard has been much developed and new needs and application scenarios have also required an important standardization effort. In this context, the IETF has considered synchronous medium access as its principal assumption for several working groups around IoT (e.g., 6lowPAN, RPL, 6tsch). Yet, regarding our proposed solutions for asynchronous MAC layer, we believe some of them would be interesting to transpose to these standardization efforts (e.g., duration of superframes and inactivity periods, allocation of guaranteed timeslots). While Cooja was not providing any satisfying implementation of IEEE 802.15.4 (beacon mode), the growing interest of multiple industry domains for the IoT is making such protocols more accessible to the research community (e.g., OpenWSN initiative).

The industrial IoT (IIoT) has motivated several dedicated standards (e.g., IEEE802.15.4-2006, IEEE802.15.4-2015) that combine Time Division Multiple Access (TDMA) with slow channel hopping in order to enable reliability and energy efficiency, while combating narrow band noise, very frequent in industrial environments. Yet, such approaches require a fine scheduling of the communications in order to avoid costly interferences and retransmissions. Numerous solutions have been recently published, thus denoting a strong interest of the research community for deterministic slow channel hopping scheduling for the IIoT.

In the frame of the co-advised PhD thesis of Rodrigo Teles (with Fabrice Théoleyre as principal advisor) [112], we have identified several new challenges that would satisfy any application scenario (e.g., joint use of hybrid and auto-adaptive strategies). The IEEE 802.15.4 standard has anticipated variable traffic by allowing nodes to book guaranteed timeslots. We are studying to what extent distributed strategies would reduce control overhead while taking into account network dynamics. For instance, node mobility is not considered yet, as industrial environments have for long been assumed as composed of static devices essentially. However, drones and mobile robots are becoming important actors of these networks and their communications with the static infrastructure should be addressed seriously. Furthermore, as shown in [113], finding the best channel to use remains a difficult problem and efficient selection in a distributed manner with low overhead (e.g., limited channel scanning) would greatly help optimize existing IIoT networks. These operations are even more important as those networks are getting more and more dynamic (e.g., node mobility, varying traffic, external interferences).

Finally, considering industrial networks and monitoring applications, a myriad of different technologies has begun to emerge on the market, along with a variety of scenarios, and using the same unlicensed band (e.g., LoRa, Sigfox). While direct links can indeed be established between end devices and gateways, thus avoiding multihop routing under harsh radio conditions, medium access is much less competitive (e.g., LoRa allows for only some bytes per end device and per day [114]). However, the coexistence of such networks with other surrounding wireless networks remains to be investigated and new mechanisms should be promoted to provide end-to-end guarantees.

5.2.3 Fault tolerance

In wireless sensor networks and IoT in general, the increasing reliability requirement imposes to compensate for the loss of messages during transmissions, as well as for nodes disappearing from the topology (e.g., fault, mobility, security issue). In [24], we had proposed to enable collaboration between nodes in order to allow for distributed data storage upon link failures. Any node losing connectivity to the sink station and reaching limits of its local storage capacity would request neighboring devices to store its data temporarily. To do so, we aimed at limiting communication overhead by overhearing communications and not requiring storing node to send data back to the initial node but rather to act as if they were their own. By using such an opportunistic first hop and then rely on the existing routing tree, we managed to avoid heavy routing reconfigurations and convergence delays that could be induced by temporary link instabilities. Jointly, we proposed a congestion control scheme at the MAC layer in order to slow data generation upon failure, based on available buffers and communication opportunities [25].

We are currently considering multiple (static or mobile) sinks whose announcements throughout the network would make decision-making more complete and relevant for outgoing data. Furthermore, nodes themselves can fail and several contributions have proposed to replace them by using mobile devices. Such substitution networks would help restoring connectivity in areas where either one critical node or multiple sensors have disappeared. Latest hardware innovations would even permit to reload batteries of some devices, thus requiring to anticipate discharge [115] (i.e., identifying nodes that should be substituted). In this context, additional control traffic and changing topologies are requiring enhanced MAC solutions. Replacement nodes could also be driven by fine-grained traffic analysis performed at MAC layer (e.g., number of retransmissions, 1-hop delay), which may help for quicker and more relevant interventions.

5.2.4 Towards a wider application of IoT protocols?

Low-power schemes for "greener" Internet

In [2], we had observed that efficient activity scheduling in WSN (i.e., allowing some devices to turn off while maintaining connected area coverage) was still leaving many multipath opportunities within the network, thus emphasizing the room for further reducing energy consumption of those networks.

The Internet seems to present similar symptoms as the increased range of available applications and communication paradigms is still far from draining its resources. As exposed in [116], end-hosts demand high availability and full-time connectivity even if the network is not used, as initially detailed in [117]. Furthermore, the expansion of data centers has forced the network infrastructure (including routers, switches etc.) to increase significantly, thus motivating research on the energy efficiency of wired networks. For instance, Elastic Tree [118] dynamically adjusts the set of active network elements (i.e., links and switches) in order to satisfy changing data center traffic loads, thus saving up to 50% of network energy, while maintaining the ability to handle traffic surges. Such approaches require local networks to let upper gateways know about their inner traffic and energy-efficiency strategies. We thus plan to investigate the mandatory collaboration between IoT networks and upper providers in order to allow each part to benefit from the energy saving policy of the other.

From routing data packets to transporting humans and goods

We have recently started a joint work around multimodal itinerary planning [119]. Our objective is to export efficient medium access control and routing methods in IoT networks to transportation networks. Similarly, once a trip is planned from one place to a destination, dynamics must be taken into account and could even be anticipated for some of them. While public transportation systems provide precise schedules of their networks, personal means are flexible (e.g., car, bike) and user requirements can be considered for each request [120]. At the european scale, the transport policy aims at a form of mobility that is sustainable, energy-efficient and respectful of the environment. These goals can be achieved by using multimodal transport that combines optimally the various modes of transport, exploiting each one's strength and minimizing the weaknesses[9].

Such initiatives will face issues that the Internet has already overcome. Similarly to computer networks, transportation networks are made of links that can become congested, hubs whose legs can appear or disappear, and so on. However, data packets and humans seem to present some differences. Several studies have indeed exposed that real-time rerouting of individuals induces negative reactions. The number of changes should thus be minimized, as well as unstable decisions [121]. In addition, multimodality requirement imposes shortest path algorithms to consider new constraints (e.g., switching times, order between transportation modes) that must be modeled accurately. We believe that several envisioned problems actually look similar to some already solved within the Internet. Interdomain routing has indeed enabled the transportation of data packets over different domains, with specific policies and constraints. But again, there might be some differences between data packets and people.

[9] https://ec.europa.eu/transport/themes/logistics_multimodal_en

Appendix A

Personal bibliography

[1] R. Kuntz, A. Gallais, and T. Noel, “Medium Access Control Facing the Reality of WSN Deployments,” *SIGCOMM Computer Communication Review*, vol. 39, no. 3, pp. 22–27, Jun. 2009.

[2] P. Merindol and A. Gallais, “Path diversity in energy-efficient wireless sensor networks,” in *Proceedings of IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2009, pp. 2280–2284.

[3] J. Beaudaux, A. Gallais, and T. Noel, “Reality-proof activity scheduling for energy-efficient wireless sensor networks,” *Inderscience International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1/2/3, pp. 185–199, 2014.

[4] J. Beaudaux, A. Gallais, and T. Razafindralambo, “Multiple coverage with controlled connectivity in wireless sensor networks,” in *Proceedings of the 7th ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2010, pp. 9–16.

[5] A. Gallais, “Ordonnancement d’activité dans les réseaux de capteurs : l’exemple de la couverture de surface.” Ph.D. dissertation, Université des Sciences et Technologies de Lille (USTL), 2007.

[6] J. Beaudaux, A. Gallais, and T. Noel, “Localized mac duty-cycling adaptations for global energy-efficiency in wireless sensor networks,” in *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, July 2013, pp. 795–800.

[7] ——, “Heterogeneous mac duty-cycling for energy-efficient internet of things deployments,” *Springer Networking Science*, vol. 3, no. 1-4, pp. 54–62, 2013.

[8] C. B. des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noel, and J. Vandaele, “Using SensLAB as a First Class Scientific Tool for Large Scale Wireless Sensor Network Experiments,” in *Proceedings of the 10th IFIP International Conference on Networking*, 2011, pp. 147–159.

[9] R. Kuntz, A. Gallais, and T. Noel, “Auto-adaptive MAC for energy-efficient burst transmissions in wireless sensor networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2011, pp. 233–238.

[10] ——, "From versatility to auto-adaptation of the medium access control in wireless sensor networks," *Elsevier Journal of Parallel and Distributed Computing (JPDC)*, vol. 71, no. 9, pp. 1236 – 1248, 2011, special Issue on Advancement of Research in Wireless Access and Mobile Systems.

[11] G. Z. Papadopoulos, J. Beaudaux, A. Gallais, and T. Noel, "T-AAD: Lightweight Traffic Auto-ADaptations for Low-power MAC Protocols," in *Proceedings of the 13th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014, pp. 79–86.

[12] G. Z. Papadopoulos, J. Beaudaux, A. Gallais, P. Chatzimisios, and T. Noel, "Toward a Packet Duplication Control for Opportunistic Routing in WSNs," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 94–99.

[13] G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noel, "Demo: Abstract: Live Adaptations of Low-power MAC Protocols," in *Proceedings of the 21st ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2015, pp. 207–209.

[14] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noel, "Wireless Medium Access Control under Mobility and Bursty Traffic Assumptions in WSNs," *Springer Mobile Networks and Applications (MONET)*, vol. 20, no. 5, pp. 649–660, 2015.

[15] G. Z. Papadopoulos, A. Gallais, T. Noel, V. Kotsiou, and P. Chatzimisios, "Enhancing ContikiMAC for Bursty Traffic in Mobile Sensor Networks," in *Proceedings of the IEEE SENSORS*, 2014, pp. 257–260.

[16] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noel, "Optimizing the Handover Delay in Mobile WSNs," in *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT)*, 2015, pp. 210–215.

[17] ——, "Low-Power Neighbor Discovery for Mobility-Aware Wireless Sensor Networks," *Elsevier Ad Hoc Networks*, vol. 48, pp. 66–79, 2016.

[18] G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noel, "Importance of Repeatable Setups for Reproducible Experimental Results in IoT," in *Proceedings of the 13th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2016.

[19] G. Z. Papadopoulos, K. Kritsis, A. Gallais, P. Chatzimisios, and T. Noel, "Performance Evaluation Methods in Ad Hoc and Wireless Sensor Networks: A Literature Study," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 122–128, 2016.

[20] J. Beaudaux, A. Gallais, R. Kuntz, J. Montavont, T. Noël, D. Roth, F. Theoleyre, and E. Valentin, "Casino: Creating alea with a sensor-based interactive network," in *Proceedings of 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2010, demo.

[21] J. Beaudaux, A. Gallais, J. Montavont, T. Noel, D. Roth, and E. Valentin, "Thorough empirical analysis of x-mac over a large scale internet of things testbed," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 383–392, Feb 2014.

[22] G. Z. Papadopoulos, J. Beaudaux, A. Gallais, T. Noel, and G. Schreiner, "Adding value to WSN simulation using the IoT-LAB experimental platform," in *Proceedings of the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013, pp. 485–490.

[23] G. Z. Papadopoulos, G. Schreiner, E. Jou, and T. Noel, “Characterizing an IoT testbed: can experiments lead to scientific results?” *Elsevier Computer Networks*, vol. 119, pp. 86–101, 2017.

[24] J. Beaudaux, A. Gallais, J. Montavont, and T. Noel, “Lift: Layer independent fault tolerance mechanism for wireless sensor networks,” in *Proceedings of IEEE 73rd Vehicular Technology Conference (VTC Spring)*, May 2011, pp. 1–5.

[25] G. Z. Papadopoulos, N. Pappas, A. Gallais, T. Noel, and V. Angelakis, “Distributed Adaptive Scheme for Reliable Data Collection in Fault Tolerant WSNs,” in *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT)*, 2015, pp. 116–121.

Appendix B

General bibliography

[26] M. Li and Y. Liu, "Underground Structure Monitoring with Wireless Sensor Networks," in *Proceedings of the 6th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 69–78.

[27] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pásztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef, "Evolution and Sustainability of a Wildlife Monitoring Sensor Network," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2010, pp. 127–140.

[28] O. Chipara, C. Lu, T. Bailey, and G. Roman, "Reliable Clinical Monitoring using Wireless Sensor Networks: Experiences in a Step-down Hospital Unit," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2010, pp. 155–168.

[29] P. Zhang, C. M. Sadler, S. A. Lyon, and M. M. Tonosi, "Hardware design experiences in zebranet," in *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2004, pp. 227–238.

[30] J. Eriksson, F. Österlind, N. Finne, A. Dunkels, N. Tsiftes, and T. Voigt, "Accurate network-scale power profiling for sensor network simulators," in *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN)*, 2009, pp. 312–326.

[31] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, ANSI/IEEE Std 802.11.

[32] M. Ali, U. Saif, A. Dunkels, T. Voigt, K. Römer, K. Langendoen, J. Polastre, and Z. A. Uzmi, "Medium access control issues in sensor networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 33–36, April 2006.

[33] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, October 2000.

[34] V. Rajendran, J. Garcia-Luna-Aveces, and K. Obraczka, "Energy-efficient, application-aware medium access for sensor networks," in *Proceedings of 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2005, pp. 630–637.

[35] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in *Proceedings of the 25th IEEE Conference on Computer Communications (INFOCOM)*, April 2006, pp. 1–13.

[36] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-Efficient, collision-free medium access control for wireless sensor networks,” *Wireless Networks*, vol. 12, pp. 63–78, February 2006.

[37] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, 2002, pp. 1567–1576.

[38] T. van Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2003, pp. 171–180.

[39] J. Pak, J. Son, and K. Han, “A MAC Protocol Using Separate Wakeup Slots for Sensor Network,” in *Computational Science and Its Applications - ICCSA 2006: International Conference, Glasgow, UK, May 8-11, 2006. Proceedings, Part II.* Springer Berlin Heidelberg, 2006, pp. 1159–1168.

[40] “IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs),” IEEE Std 802.15.4e-2012 (Revision of IEEE Std 802.15.4-2011), IEEE, April 2012.

[41] A. El-Hoiydi, “Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks,” in *Proceedings of IEEE International Conference on Communications (ICC)*, vol. 5, 2002, pp. 3418–3423 vol.5.

[42] J. L. Hill and D. E. Culler, “Mica: a wireless platform for deeply embedded networks,” *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov 2002.

[43] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks,” in *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2004, pp. 95–107.

[44] I. Rhee, A. Warrier, M. Aia, J. Min, and M. Sichitiu, “Z-MAC: A Hybrid MAC for Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, June 2008.

[45] W. Ye, F. Silva and J. Heidemann, “Ultra-Low Duty Cycle MAC with Scheduled Channel Polling,” in *Proceedings of 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006, pp. 321–334.

[46] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, “A Wireless Sensor MAC Protocol for Bursty Data Traffic,” in *Proceedings of the 18th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, September 2007, pp. 1–5.

[47] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, “Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks,” in *Proceedings of 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006, pp. 293–306.

[48] K. Kredo, II and P. Mohapatra, “Medium Access Control in Wireless Sensor Networks,” *Elsevier Computer Networks*, vol. 51, no. 4, pp. 961–994, 2007.

[49] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in WSNs: A survey of preamble sampling MAC protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351–3363, 2011.

[50] ——, "Low energy operation in WSNs: A survey of preamble sampling MAC protocols," *Comput. Networks*, vol. 55, no. 15, pp. 3351–3363, 2011.

[51] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceedings of 1st International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002, pp. 88–97.

[52] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Out-of-the-box environmental monitoring," in *Proceedings of the 7th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* ACM/IEEE, April 2008, pp. 332–343.

[53] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2007, pp. 254–263.

[54] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "SensorFlock: an Airborne Wireless Sensor Network of Micro-air Vehicles," in *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, November 2007, pp. 117–129.

[55] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: Enabling mobility in sensor networks," in *Proceedings of the 4th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2005, pp. 404–409.

[56] P. Dutta, D. Culler, and S. Shenker, "Procrastination Might Lead to a Longer and More Useful Life," in *Proceedings of Sixth Workshop on Hot Topics in Networks (HotNets-VI)*, 2007.

[57] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks," in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2006.

[58] C. J. Merlin and W. B. Heinzelman, "Network-aware adaptation of MAC scheduling for Wireless Sensor Networks," in *Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2007.

[59] K.-J. Wong and D. K. Arvind, "SpeckMAC: Low-power Decentralised MAC Protocols for Low Data Rate Transmissions in Specknets," in *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, 2006, pp. 71–78.

[60] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," *Swedish Institute of Computer Science, Technical Report*, 2011.

[61] D. Moss and P. Levis, "BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking," *Technical Report SING-08-00, Stanford University*, 2008.

[62] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, 2007.

[63] H. Yoo, M. Shim, and D. Kim, “Dynamic duty-cycle scheduling schemes for energy-harvesting wireless sensor networks,” in *IEEE Communications Letters*, vol. 16, 2012, pp. 202–204.

[64] A. El-Hoiydi and J.-D. Decotignie, “WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks,” in *Proceedings of First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, 2004, pp. 18–31.

[65] X. Wang, X. Wang, G. Xing, and Y. Yao, “Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks,” in *Proceedings of IEEE 18th International Workshop on Quality of Service (IWQoS)*, June 2010, pp. 1–9.

[66] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, “pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols,” in *Proceedings of the 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 173–184.

[67] X.-Y. Li, Y. Wang, and W.-Z. Song, “Applications of k-local MST for topology control and broadcasting in wireless ad hoc networks,” *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 15, no. 12, pp. 1057–1069, 2004.

[68] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” Internet Engineering Task Force Request for Comments (RFC) 6550, 2012.

[69] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Transactions on Networking (ToN)*, vol. 11, no. 1, pp. 2–16, 2003.

[70] F. Khadar and T. Razafindralambo, “Performance evaluation of gradient routing strategies for wireless sensor networks,” in *Proceedings of IFIP Networking*, 2009, pp. 535–547.

[71] A. Fraboulet, G. Chelius, and E. Fleury, “Worldsens: Development and Prototyping Tools for Application Specific Wireless Sensors Networks,” in *Proceedings of International Symposium on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 176–185.

[72] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks,” in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2006, pp. 307–320.

[73] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, “FIT IoT-LAB: A large scale open experimental IoT testbed,” in *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT)*, 2015, pp. 459–464.

[74] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, “FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments,” in *Proceedings of International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2006, pp. 28–41.

[75] J. Leguay, M. Lopez-Ramos, K. Jean-Marie, and V. Conan, “An Efficient Service Oriented Architecture for Heterogeneous and Dynamic Wireless Sensor Networks,” in *Proceedings of 3rd IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SensApp)*, 2008.

[76] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, “Deploying a Wireless Sensor Network on an Active Volcano,” *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, March/April 2006.

[77] M. Ceriotti, L. Mottola, G. Picco, A. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, “Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment,” in *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, April 2009, pp. 277–288.

[78] C. Merlin and W. Heinzelman, “Duty cycle control for low-power-listening MAC protocols,” in *Proceedings of 5th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2008, pp. 497–502.

[79] C. J. Merlin and W. B. Heinzelman, “Schedule Adaptation of Low-Power-Listening Protocols for Wireless Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 672–685, 2010.

[80] M. Anwander, G. Wagenknecht, T. Braun, and K. Dolfus, “BEAM: A Burst-aware Energy-efficient Adaptive MAC protocol for Wireless Sensor Networks,” in *Proceedings of Seventh International Conference on Networked Sensing Systems (INSS)*, 2010, pp. 195 –202.

[81] G. Corbellini, E. Strinati, and A. Duda, “LA-MAC: Low-latency asynchronous MAC for wireless sensor networks,” in *Proceedings of 23rd annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2012, pp. 380–386.

[82] C. Merlin and W. Heinzelman, “Duty Cycle Control for Low-Power-Listening MAC Protocols,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1508 –1521, 2010.

[83] P. Hurni and T. Braun, “MaxMAC: a maximally traffic-adaptive MAC protocol for wireless sensor networks,” in *Proceedings of the 7th European conference on Wireless Sensor Networks (EWSN)*, 2010, pp. 289–305.

[84] H. Liu, B. Zhang, H. T. Mouftah, X. Shen, and J. Ma, “Opportunistic Routing for Wireless Ad Hoc and Sensor Networks: Present and Future Directions,” *IEEE Communications Magazine*, vol. 47, no. 12, pp. 103–109, 2009.

[85] P. Spachos, P. Chatzimisios, and D. Hatzinakos, “Energy Aware Opportunistic Routing in Wireless Sensor Networks,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 405–409.

[86] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, “Low Power, Low Delay: Opportunistic Routing meets Duty Cycling,” in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 185–196.

[87] S. Duquennoy, O. Landsiedel, and T. Voigt, “Let the Tree Bloom: Scalable Opportunistic Routing with ORPL,” in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2013, pp. 1–14.

[88] L. Mokdad, J. Ben-Othman, B. Yahya, and S. Niagne, “Performance Evaluation Tools for QoS MAC Protocol for Wireless Sensor Networks,” *Ad Hoc Networks*, vol. 12, pp. 86–99, 2014.

[89] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-Level Sensor Network Simulation with COOJA,” in *Proceedings of the 31st Annual IEEE International Conference on Local Computer Networks (LCN)*, 2006, pp. 641–648.

[90] J. H. Ahn and T. J. Lee, “Allys: All you can send for energy harvesting networks,” *IEEE Transactions on Mobile Computing*, August 2017, to appear.

[91] W. Dargie and J. Wen, “A Seamless Handover for WSN Using LMS Filter,” in *Proceedings of the 39th IEEE Conference on Local Computer Networks (LCN)*, 2014, pp. 287–288.

[92] T. Watteyne, M. Palattella, and L. Grieco, “Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement,” RFC 7554, 2015.

[93] A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux, “WiseMAC, an Ultra Low Power MAC Protocol for the wiseNET Wireless Sensor Networks,” in *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2003, pp. 302–303.

[94] S. Duquennoy, F. Österlind, and A. Dunkels, “Lossy Links, Low Power, High Throughput,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2011, pp. 12–25.

[95] G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke, “Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 63–72, Jan 2014.

[96] Q. Dong and W. Dargie, “A Survey on Mobility and Mobility-Aware MAC Protocols in Wireless Sensor Networks,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 88–100, 2013.

[97] T. Zhiyong and W. Dargie, “A Mobility-Aware Medium Access Control Protocol for Wireless Sensor Networks,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM) Workshops*, 2010, pp. 109–114.

[98] K. Heurtefeux and F. Valois, “Distributed Localization Protocol for Routing in a Noisy Wireless Sensor Network,” in *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2009, pp. 223–230.

[99] Q. Dong and W. Dargie, “Analysis of the Cost of Handover in a Mobile Wireless Sensor Network,” in *Proceedings of the 10th International Conference on Wired/Wireless Internet Communication (WWIC)*, 2012, pp. 315–322.

[100] Y. Sun, O. Gurewitz, and D. 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 Networked Sensor Systems (Sensys)*, 2008, pp. 1–14.

[101] W. Dargie and J. Wen, “A MAC Protocol for Medical Applications (Poster Abstract),” in *Proceedings of the 13th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2014, pp. 287–288.

[102] R. Kuntz, J. Montavont, and T. Noël, "Improving the medium access in highly mobile Wireless Sensor Networks," *Telecommunication Systems*, vol. 52, no. 4, pp. 2437–2458, 2013.

[103] P. D. Ba, I. Niang, and B. Gueye, "An Optimized and Power Savings Protocol for Mobility Energy-aware in Wireless Sensor Networks," *Telecommunications Systems*, vol. 55, no. 2, pp. 271–280, 2014.

[104] D. Roth, J. Montavont, and T. Noel, "MOBINET: Mobility management across different wireless sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2011, pp. 351–356.

[105] C. Dandelski, B.-L. Wenning, D. Perez, D. Pesch, and J.-P. Linnartz, "Scalability of dense wireless lighting control networks," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 157–165, January 2015.

[106] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A Mobility Scenario Generation and Analysis Tool," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools)*, 2010, pp. 51:1–51:10.

[107] B. Dezfouli, M. Radi, and O. Chipara, "REWIMO: A Real-Time and Reliable Low-Power Wireless Mobile Network," *ACM Transactions on Sensor Networks*, vol. 13, no. 3, pp. 17:1–17:42, August 2017.

[108] Y. Liu, H. Liu, Q. Yang, and S. Wu, "RM-MAC: A routing-enhanced multi-channel MAC protocol in duty-cycle sensor networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, June 2015, pp. 3534–3539.

[109] M. Bezunartea, M. Gamallo, J. Tiberghien, and K. Steenhaut, "How Interactions between RPL and Radio Duty Cycling Protocols Affect QoS in Wireless Sensor Networks," in *Proceedings of the 12th ACM Symposium on QoS and Security for Wireless and Mobile Networks, (Q2SWinet@MSWiM)*, 2016, pp. 135–138.

[110] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has Time Come to Switch from Duty-cycled Mac Protocols to Wake-up Radio for Wireless Sensor Networks?" *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 674–687, April 2016.

[111] A.-S. Tonneau, N. Mitton, and J. Vandaele, "How to choose an experimentation platform for wireless sensor networks? a survey on static and mobile wireless sensor network experimentation facilities," *Elsevier Ad Hoc Networks*, vol. 30, pp. 115 – 127, 2015.

[112] R. T. Hermeto, "Virtualizing Heterogeneous Wireless Networks with SDN for the 5G," Ph.D. dissertation, University of Strasbourg, in preparation, funded by IdEx program.

[113] V. Toldov, L. Clavier, V. Loscrí, and N. Mitton, "A Thompson Sampling Approach to Channel Exploration-Exploitation Problem in Multihop Cognitive Radio Networks," in *Proceedings of 27th annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2016, pp. 1–6.

[114] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," in *Proceedings of 22th European Wireless Conference (EW)*, 2016, pp. 1–6.

[115] L. Militano, M. Erdelj, A. Molinaro, N. Mitton, and A. Iera, "Recharging vs. Replacing Sensor Nodes Using Mobile Robots for Network Maintenance," *Telecommunications Systems*, February 2016.

[116] T. Haudebourg and A.-C. Orgerie, "On the energy efficiency of sleeping and rate adaptation for network devices," in *Proceedings of the 17th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, 2017, pp. 132–146.

[117] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (alr)," *IEEE Transactions on Computers*, vol. 57, no. 4, pp. 448–461, 2008.

[118] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI)*. USENIX Association, 2010, pp. 17–17.

[119] M. A. Falek, "Optimized Algorithms for Customized Planification of Dynamic and Multimodal Itineraries," Ph.D. dissertation, University of Strasbourg, in preparation, funded by CIFRE grant with Technology & Strategy.

[120] J. Dibbelt, T. Pajor, and D. Wagner, "User-Constrained Multimodal Route Planning," *Journal of Experimental Algorithmics*, vol. 19, pp. 1.1–1.19, Apr. 2015.

[121] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "A tabu search algorithm for rerouting trains during rail operations," *Transportation Research Part B: Methodological*, vol. 44, no. 1, pp. 175 – 192, 2010.

Résumé

Cette thèse d'habilitation à diriger des recherches porte sur l'accès au medium (MAC) dans l'Internet des objets (IoT, consistant en des réseaux maillés, sans fils et multi-sauts). Face à des protocoles synchronisés convenant à des scénarios spécifiques (modèles de transmissions connus a priori, topologies statiques), j'ai étudié des approches asynchrones, mettant en oeuvre des mécanismes d'écoute basse consommation (LPL pour *low-power listening*), dans des réseaux dynamiques (trafic variable, mobilité, pannes). Depuis 2008, j'ai eu le plaisir de collaborer avec plusieurs personnes sur ces sujets, notamment des doctorants. Ce manuscrit détaille nos recherches et les résultats auxquels nous sommes parvenus en proposant des solutions cohérentes et performantes, validées par simulations et expérimentations à grande échelle. Au cours de cette période, de nombreux protocoles ont été développés au sein de la communauté, visant la configuration et l'adaptation du contrôle d'accès au medium dans des environnements mêlant variabilité du trafic de données et mobilité des noeuds. Nous sommes parvenus à des objets capables de configurer et adapter automatiquement leurs paramètres LPL dans de tels réseaux dynamiques, tout en respectant des contraintes fortes (processus décentralisés, robustesse, passage à l'échelle). Cette habilitation détaille l'essentiel de nos contributions. Elle présente également plusieurs perspectives dans la suite de ces recherches, allant de l'Internet des objets industriel à la mobilité des données, en incluant l'efficacité énergétique d'un Internet en mutation permanente.

Mots-clés: Internet des objets, réseaux maillés, sans fils et multi-sauts, contrôle de l'accès au medium, écoute basse consommation, auto-configuration, auto-adaptation.

Abstract

This thesis addresses medium access control (MAC) in the Internet of Things (IoT, consisting in the interconnection of wireless mesh and multihop networks). While synchronized time-division based protocols are suitable to some specific scenarios (a priori known traffic, static topologies), we have investigated asynchronous mechanisms that rely on low-power listening (LPL) and thus allow to consider dynamic wireless networks (variable traffic, mobility, faults). Since 2008, I have had the pleasure to work on this subject with several people, especially PhD students. This manuscript presents our research process along with the main coherent and performing solutions that we could validate through simulations and large-scale experiments. During this period, a myriad of protocols have been developed within our community, targeting configuration and adaptation of medium access control in environments prone to variable traffic and node mobility. We enabled self-configuration and self-adaptation of LPL parameters for objects involved in such dynamic networks, while proposing decentralized, robust and scalable mechanisms. This thesis also introduces several perspectives, ranging from Industrial Internet of Things (IIoT) to data mobility, while including energy efficiency of a constantly evolving Internet.

Keywords: Internet of Things, wireless multihop mesh networks, medium access control, low power listening, self-configuration, self-adaptation.