Efficient algorithms for improving packet routing and congestion control in dense electromagnetic nanonetworks

Ali Medlej

THÈSE DE DOCTORAT

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE-FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

ALI MEDLEJ

Efficient algorithms for improving packet routing and congestion control in dense electromagnetic nanonetworks
Algorithmes efficaces pour améliorer le routage des paquets et le contrôle de la congestion dans les nanoréseaux électromagnétiques denses

Thèse présentée et soutenue à Montbéliard, le 08 December 2022

Composition du Jury :

THEOLEYRE FABRICE	Directeur de Recherche, CNRS, Université de Strasbourg	Président du jury
GUITTON ALEXANDRE	Professeur, Université de Clermont Auvergne	Rapporteur
RONDEAU ERIC	Professeur, Université de Lorraine	Rapporteur
DEDU EUGEN	Maître de conférences HDR, Université de Franche-Comté	Directeur de thèse
DHOUTAUT DOMINIQUE	Maître de conférences, Université de Franche-Comté	Codirecteur de thèse
BEYDOUN KAMAL	Maître de conférences, Université Libanaise	Codirecteur de thèse

N° | X | X | X |

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis director, Prof. Eugen Dedu, Assistant professor HDR at Université Franche-Comté, for the constant support and guidance throughout the entire Ph.D. and he was my guidance to steadily advance towards the successful completion of this thesis.

I would like to express my gratitude to my thesis co-directors, Dr. Dominique Dhoutaut, Assistant Professor at Université Franche-Comté, and Dr. Kamal Beydoun Assistant Professor at Université Libanaise for their attention to all my works and their encouragement to finish the thesis. I took great pleasure to work with them.

I would like to extend my sincere thanks to the members of the committee who kindly agreed to serve in my Ph.D. defense committee and to their valuable remarks on my dissertation.

A special thanks to all the reviewers who took the time to correct this manuscript.

Finally, I would like to express my deepest love to my family for their encouragement, support and confidence. I would also like to especially thank my fiancée, Eva Zalzali, for her constant support, and my urging to complete my Ph.D.

CONTENTS

I

CONTEXT

SCIENTIFIC AND INSTITUTIONAL CONTEXT

I followed my thesis at the FEMTO-ST institute, Université Bourgogne Franche-Comté, in Montbéliard, France. I worked in the DISC department within the team OMNI (Optimisations, Mobility, Networking), and I was supervised by Prof. Eugen Dedu and co-supervised by Dr. Dominique Dhoutaut and Dr. Kamal Beydoun. This thesis took place from October 1, 2019 to September 2022. The majority of the time, I worked in Lebanon in L'aricod laboratory at the Lebanese university. I also visited FEMTO-ST laboratory at Montbéliard, France, where I worked under the supervision of Prof. Eugen for a period of 50 days.

The nanonetwork field is well known by the OMNI team. Several researches in nanonetwork subjects took place and vary between studying the coding of information, transmission of images and videos, developing a nanonetwork simulator. In addition, programmable matter and other types of applications based on nanorobots have also been explored.

My own thesis therefore fits into these existing themes in considering the network and telecommunication aspect offered by nanonetworks.

During the PhD the following articles were published or submitted

Publication in journals:

- Ali Medlej and Eugen Dedu and Kamal Beydoun and Dominique Dhoutaut, "Self-configuring asynchronous sleeping in heterogeneous networks", in ITU Journal on Future and Evolving Technologies, volume 2 (2021), Issue 7 - Terahertz communications, pp. 51-62.

Publication in conferences:

- Ali Medlej and Kamal Beydoun and Eugen Dedu and Dominique Dhoutaut, "Scaling up Routing in Nanonetworks with Asynchronous Node Sleeping", in 28th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Hvar, Croatia, pp. 177–182, Sep. 2020, IEEE, CORE B.
- Ali Medlej and Eugen Dedu and Dominique Dhoutaut and Kamal Beydoun, "Efficient Retransmission Algorithm for Ensuring Packet Delivery to Sleeping Destination Node", in International Conference on Advanced Information Networking and Applications (AINA), Sydney, Australia, pp. 219-230, Apr, 2022, Springer, CORE B.
- Ali Medlej and Dominique Dhoutaut and Kamal Beydoun and Eugen Dedu, "Reducing Network Congestion by Splitting Traffic Over Multiple Paths", in The 38th

ACM/SIGAPP Symposium On Applied Computing (SAC), 8 pages, IEEE, CORE B.

1

INTRODUCTION

The prefix ”nano” is referring to a Greek prefix meaning ”dwarf” or something very small and depicts one thousand millionth of a meter (10^{-9}m). So, the simplest definition of nanotechnology is ”technology on the nanoscale”. We should distinguish between nanoscience, and nanotechnology. Nanoscience is the study of structures and molecules on the scales of nanometers ranging between 1 and 100 nm, whereas the technology that utilizes it in practical applications such as devices etc. is called nanotechnology. As a comparison, a single human hair is 60,000 nm thickness and the DNA double helix has a radius of 1 nm Fig. 1.1. The development of nanoscience can be traced to the time of the Greeks and Democritus in the 5th century B.C., when scientists considered the question of whether matter is continuous, and thus infinitely divisible into smaller pieces, or composed of small, indivisible and indestructible particles, which scientists now call atoms.

Nanotechnology is one of the most promising technologies of the 21st century. It is the ability to convert the nanoscience theory to useful applications by observing, measuring,

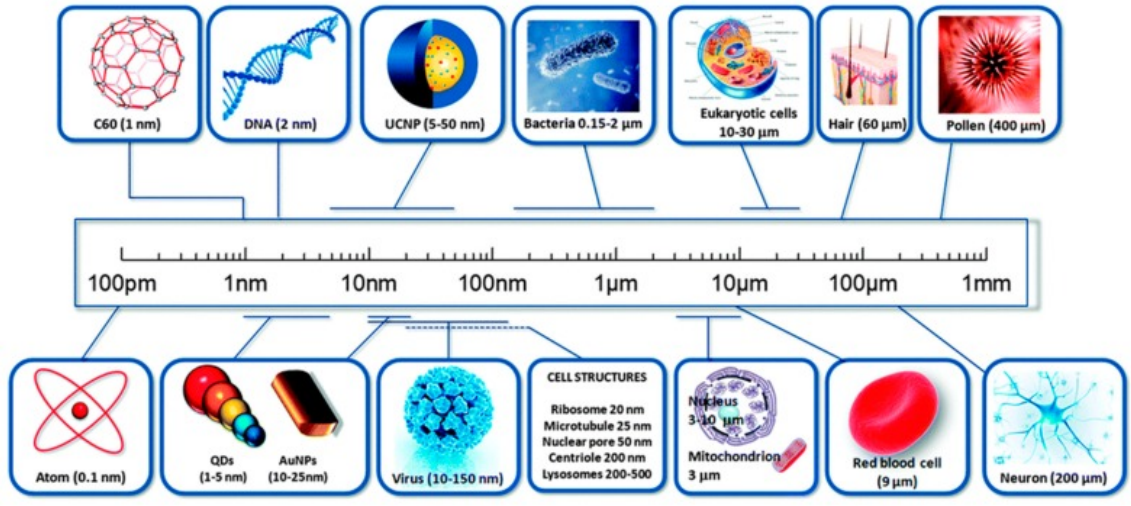


Figure 1.1: Nanomaterial sizes. https://www.mdpi.com/1420-3049/25/1/112/htm

manipulating, assembling, controlling and manufacturing matter at the nanometer scale. The National Nanotechnology Initiative (NNI) in the United States defines nanotechnology as “a science, engineering, and technology conducted at the nanoscale (1 to 100 nm), where unique phenomena enable novel applications in a wide range of fields, from chemistry, physics and biology, to medicine, engineering and electronics”. This definition suggests the presence of two conditions for nanotechnology. The first is an issue of scale: nanotechnology is concerned to use structures by controlling their shape and size at nanometer scale. The second issue has to do with novelty: nanotechnology must deal with small things in a way that takes advantage of some properties because of the nanoscale Allhoff (2009).

1.1/ THE HISTORY OF NANOTECHNOLOGY

The American physicist and Nobel Prize laureate Richard Feynman introduced the concept of nanotechnology in 1959. During the annual meeting of the American Physical Society, Feynman presented a lecture entitled ”There’s Plenty of Room at the Bottom” at the California Institute of Technology (Caltech). In this lecture, Feynman made the hypothesis ”Why can’t we write the entire 24 volumes of the Encyclopedia Britannica on the head of a pin?”, and described a vision of using machines to construct smaller machines and down to the molecular level Feynman and Machinery (1960). This new idea demonstrated that Feynman’s hypotheses have been proven correct, and for these reasons, he is considered the father of modern nanotechnology. After fifteen years in 1974, Norio Taniguchi, a Japanese scientist, was the first to use and define the term “nanotechnology” as: ”nanotechnology mainly consists of the processing of separation, consolidation, and deformation of materials by one atom or one molecule” Taniguchi (1974).

After Feynman presented this new field of research catching the interest of many scientists, two approaches have been developed describing the different possibilities for the synthesis of nanostructures. These manufacturing approaches fall under two categories: top-down and bottom-up, which differ in degrees of quality, speed and cost.

In 1986, K. Eric Drexler published the first book on nanotechnology ”Engines of Creation: The Coming Era of Nanotechnology”, which made the theory of ”molecular engineering” becoming more popular. Drexler described the build-up of complex machines from individual atoms, which can independently manipulate molecules and atoms and thereby produce self-assembly nanostructures. Later on, in 1991, Drexler, Peterson and Pergamit published another book entitled ”Unbounding the Future: the Nanotechnology Revolution” in which they use the terms ”nanobots” or ”assemblers” for nano processes in medicine applications and then the classical term ”nanomedicine” was used for the first time after that Drexler et al. (1991).

1.2/ NANONETWORKS

We define nanodes as nanomachines that have very low hardware capabilities and cannot completely be used individually. Those are able to perform simple and basic tasks, for example, data storing, computing, actuation, and sensing. A set of interconnected nanomachines (devices a few hundred nanometres) is known as nanonetwork. This technology is also called by the name nanoscale network. In nanonetworks, the channel capacity is very high (in Tb/s) in THz band Jornet and Akyildiz (2010).

Due to their very small size, the nanonodes suffer from their modest capacity (low battery, low memory, low CPU), which negatively affects their full performance. Thus, the technical capacity of the nodes must be taken into account in each simulated scenario.

Nanonetworks are relied upon to grow the capacities of single nanomachines in both the range of operation and complexity by permitting them to share, coordinate and fuse the data or information. Nanonetworks empower new uses of nanotechnology in various different fields such as environmental research, biomedical field, industrial and military technology.

Different approaches in the literature focus on designing and developing routing protocols for heterogeneous electromagnetic nanonetworks. Nowadays, the nano-scale communication technology enters into many important sectors such as health, military, agriculture, programmable matter, etc.

Programmable matter We define programmable matter as a matter which has the ability to change its physical properties (shape, density, conductivity, etc.) in a programmable fashion, based upon user input or autonomous sensing, as shown on Fig. 1.2.

In 1991, the term programmable matter is promoted. It refers to an ensemble of fine-grained computing elements arranged in space Toffoli and Margolus (1991). In this paper, authors describes a computing substrate that is composed of fine-grained compute nodes distributed throughout space which communicate using only nearest neighbor interactions.

Nanotechnology, and self-replicating machine technology have advanced, the use of the term programmable matter has changed to reflect the fact that it is possible to build an ensemble of elements which can be ”programmed” to change their physical properties in reality, not just in simulation. Other way to say, programmable matter has come to mean ”any bulk substance which can be programmed to change its physical properties.”

Computer science To keep pace with the constant miniaturization of computer chips, transistors must have increasingly smaller features. Unfortunately, according to experts,

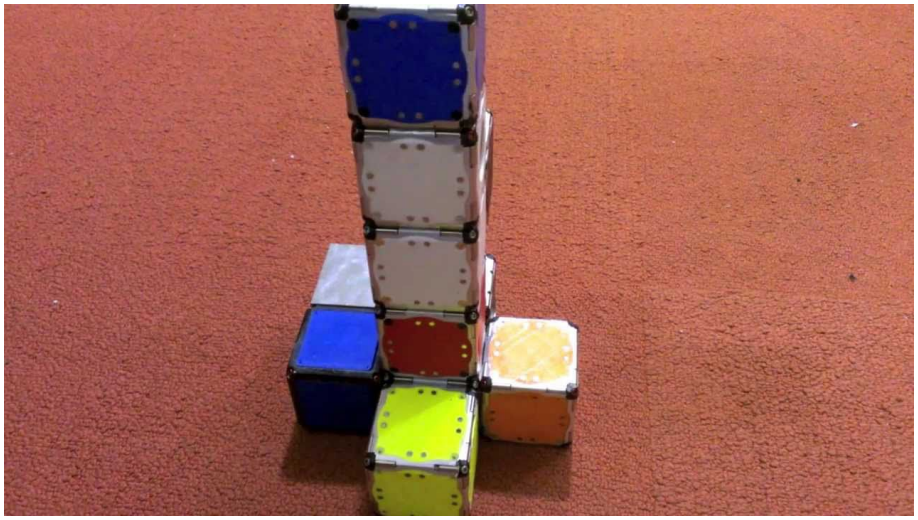


Figure 1.2: Make your own world with programmable matter. https://spectrum.ieee.org/new-ieee-life-members-website

silicon begins to run out of steam at around five nanometers. A nanoscale semiconductor with advantageous electronic qualities might be used to make transistors in next-generation computers.

Carbon nanotubes Ghadiry et al. (2012) is one technology that looks able to maintain the current pace of technological advancement by providing a sensible approach to smaller, faster transistors. IBM is one major player planning to have transistors built using carbon nanotubes that are able to take control from silicon transistors soon after 2020, when silicon-based transistors are anticipated to reach their limit.

IBM's design makes use of six nanotubes arranged in parallel to create a single transistor. Each nanotube is 1.4 nanometers wide, about 30 nanometers long, and spaced approximately eight nanometers apart from its neighbors. Both ends of the tubes are rooted in electrodes that provide current, leaving about 10 nanometers of their lengths uncovered in the middle. A third electrode extends perpendicularly underneath this part of the tubes and switches the transistor on and off to signify digital 1's and 0's.

Health sector Heterogeneous networks pervade in many applications where nodes are distributed depending on the environment conditions. Due to their nanoscale size, nanonodes can be deployed at different scales Ali and Abu-Elkheir (2015) inside the human body and across a variety of environmental contexts Kim et al. (2010). Nanonode uses are not limited to sensing and monitoring human vital signs, but it may cover some other operations when needed Stelzner et al. (2016).

An example of a medical application that can be assigned to nanonodes and needs high reliability and accuracy is drug delivery Nichols and Bae (2012) (Fig. 1.3). Heterogeneity

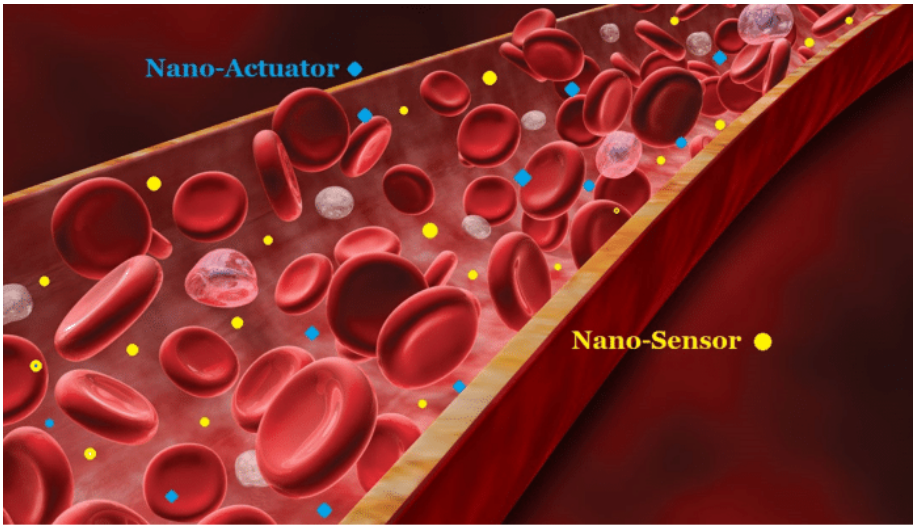


Figure 1.3: Drug delivery in human body. https://www.semanticscholar.org/paper/A-Comprehensive-Survey-on-Hybrid-Communication-for-Yang-Bi/27ef6cc2433f504e48cf760efd2dc3959ac38630

resides in the human body through the variation in the size and shape of the human organs Maksimović (2017). Nanomedicine may be defined as the use of nano-devices and nanostructures for monitoring, repairing or controlling human biological systems at the molecular level Seyedi et al. (2013). The nanonodes spreading in blood vessels can monitor the glucose level, and at the same time release the insulin to regulate the glucose level. The risk lies when the patient forgets to take his medicine especially those related to the control of certain levels of enzymes in the blood (as in the example above). With this type of technology, the patient becomes more able to control his health troubles.

This heterogeneity of the human body poses a major challenge to the routing protocol by traversing areas of different densities.

Agriculture sector Agriculture is an important source of livelihood in most parts of the world. Wireless nano-sensor(s) have been used in modern agriculture and farming (Precision Agriculture (PA)), defined as the techniques of applying farming parameters and resources for production optimization, and reducing human effort Lee et al. (2010).

To improve productivity (e.g. detection of plant viruses, soil humidity degree), a farmer needs to monitor numerous parameters (Fig. 1.4). Wireless nano-sensors need to be used to realize the vision of intelligent agriculture. One can deploy autonomous nano-sensors around the plants to monitor soil condition and plant viruses, which gives them all the information about the environmental conditions of the plants using a simple portable device.

WNSN (Wireless Nano-Sensor Network) technology will contribute in generating the tools

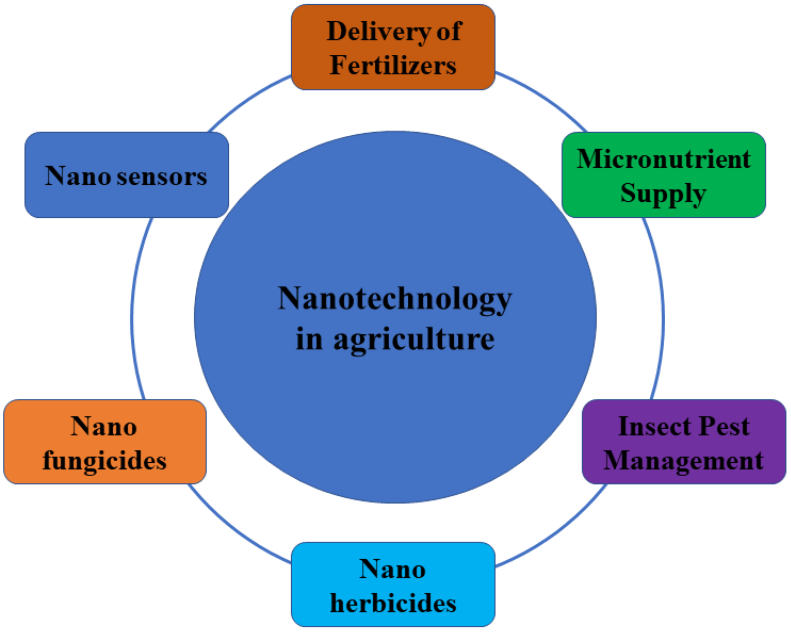


Figure 1.4: Nanonetwork technology impact in agriculture.

for establishing a real-time plant monitoring system, composed of a chemical nano-sensor merged with plants. Chemical nano-sensor nodes are miniaturized machines that interact with the environment to collect chemical compounds disseminated by plants. Micro-gateways can interconnect data collected from the nanonetwork to the external network, and to the decision officer of the analytical laboratory Giraldo et al. (2019).

Civil engineering sector Developments in nanotechnology can benefit construction engineering by enabling the practical deployment of structural condition monitoring systems for large civil engineering systems (Fig. 1.5). Nanonodes can be integrated into a composite material that can provide information on its performance and environmental conditions by monitoring structural loads, temperature, and humidity. It can be used for the construction of smart buildings Wegner et al. (2006). Also, they could be coated into bridges, as prevention monitoring against degradation and cracking in order to alert the decision-making authorities well before the damage is detectable by human inspectors Zheng et al. (2011).

Nanonodes could also be embedded into roads and structures to allow engineers to monitor deterioration and cracking, thus saving physical intervention. Road sensor networks could gather and provide data to transport operators to better control and detect congestion and incidents.

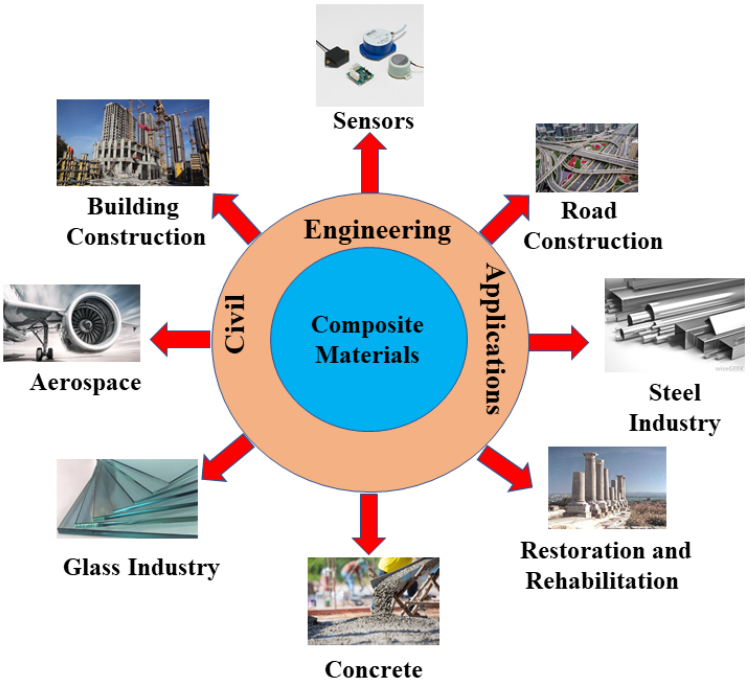


Figure 1.5: Smart composite materials based on nanotechnology, for civil engineering applications.

Environment sector Environmental safety is one of the sectors where nanotechnology may have a major impact. Through the installation of nano-sensors in high density public locations (e.g. hospitals, airports, and restaurants), authorities can trace the circulation of viruses and notice how various types of people are affected. Nanosensor networks could also be utilised to monitor the environment, such as pollution and gas emission.

Water is the lifeblood of the world, hence the importance of monitoring its quality and safety. We can benefit from nano-sensors in detecting bacteria, diseases and other harmful infectious agents, especially when improvement is made close to the point of use at the household. Nano-sensors can also be used in flood-prone rivers near residential places. These sensors can measure the water level in rivers, and send the information to the control and decision room in real time (Fig. 1.6).

To summarise, numerous applications use heterogeneous wireless networks. The biggest challenges remain in two factors, firstly the design of the routing protocols, which will face the challenges of different node densities in several regions, and secondly the node's hardware constraints, especially the small energy storage capacity and its inability to replace the battery. Therefore, thinking to preserve resources (especially energy) might be a solution to extend the network and nodes' lifetime. In this context, we focus on the influence of the sleep mechanism on the delivery of data packets to their desired destination, as well as contributing to preserving the nodes' resources.

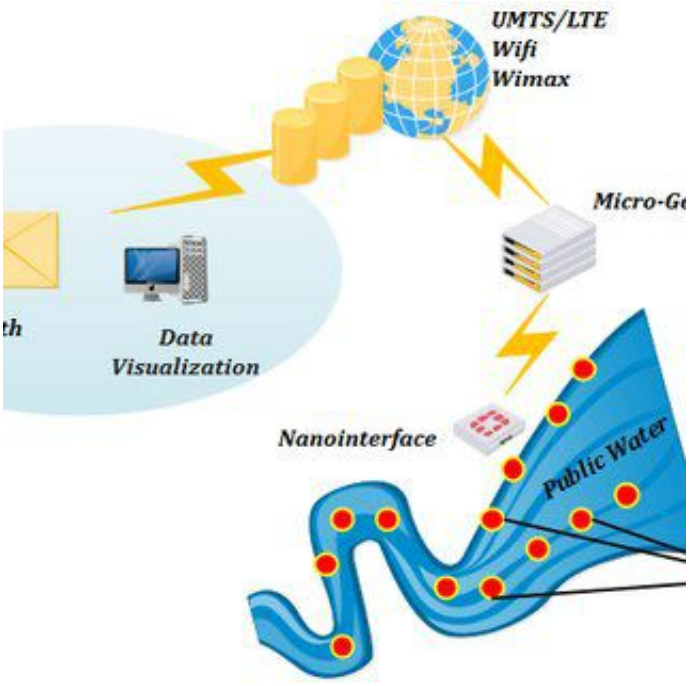


Figure 1.6: A water monitoring architecture. https://www.researchgate.net/profile/Driss-El-Ouadghiri/publication/348355577/figure/fig5/AS:978070009815043@1610201470250/Water-quality-monitoring-architecture-W640.jpg

1.3/ CONTRIBUTION TO NANONETWORK

Nanonetwork field is very young. Most of the researches focused on the study of the transport layer, as well as the physical layer. The transport layer and its most important part, congestion control, have not yet been studied. Note that network and transport layers are sometimes linked in an approach called cross-layer, which shows the interest in current collaborations. We are interested in the routing layer as well as the transport layer.

Our contribution reported in this thesis dissertation concerns proposals for new algorithms dealing with improving packet routing in a way to help preserve the node's resources and improve the treatment of network congestion among dense nanodevices. We hope that our study will open the door to deeply exploring the transport layer, which will lead to improving inter-nodes communication.

The constitution of a base of knowledge and tools relevant to the study of nanonetworks has been a common thread throughout this thesis work.

Since there is no possibility to make a real scenario due to the fact that nanonodes have not been manufactured yet, it is necessary to use a simulator as a relative to simulate scenarios, analyze results and extract conclusions. A dedicated chapter 3 describes in detail the simulator software we have used with its related features and capabilities.

The contribution part presents the work carried out during this thesis. Those are organized as follows:

- Duty-cycling in homogeneous networks: A common technique to preserve node resources (especially energy consumption) is to use duty cycling (sleeping) techniques, where nodes wake up from time to time to receive packets sent to them. Our proposed sleeping mechanism differs from those used in macro-scale network on two main aspects (fine granularity, asynchronism). 4.4.1 explains in detail the effectiveness of our proposed algorithm on preserving nodes resources.
- Duty-cycling in heterogeneous networks with destination zone: Represents the implementation and testing of the sleeping technique in a heterogeneous network. In addition, the awaken duration for the nodes will be dynamically determined based on a density estimator algorithm. A packet retransmission algorithm has been proposed as a solution for packet loss, when a packet arrives at the destination zone and the destination node is asleep. 4.3.2.1 explain in details the proposed algorithm, while the evaluation results are presented in 4.4.2.
- Probabilistic packet retransmission at the destination zone: Making all nodes at the destination zone to retransmit the packet leads to nodes' resources being exhausted. To avoid this problem, we propose a probabilistic retransmission algorithm, where not all the nodes participate in the retransmission mechanism. 4.3.2.2 explains in detail the enhanced retransmission algorithm and it's effectiveness on decreasing congestion chance at the destination zone, while the evaluation results are presented in 4.4.3.

Because of the chronological order of my PhD thesis, I started by finding a solution to preserve nodes resources by decreasing the number of forwarders during packet routing. The proposed sleeping technique showed its effectiveness in decreasing node resource consumption. On the other hand, we found another problem that affects the reliability of packet receptions at the destination zone.

When the packet arrive at the destination, and if the destination node is asleep, then the packet will be lost. For that reason, we also proposed a packet retransmission algorithm with its enhancement to increase the chance of packet reception at the destination zone. This algorithm relies on retransmitting the packet by each node at the end of its awaken duration at the destination zone. By doing so, we are increasing the reliability of packet receptions.

Allowing all nodes to participate in packet retransmission at the destination zone might lead to increase congestion phenomena in this area, as well as exhausting nodes' resources over time. Hence, we proposed an enhancement for the full retransmission algo-

rithm. In the enhanced solution, we aim to decrease the number of participating nodes in packet retransmission, while guaranteeing the packet reception by its intended node.

Recall that applying the sleeping mechanism alone shows its effectiveness in reducing the number of forwarder nodes along the routing way from the source to the destination. To outcome the perfect result from applying the sleeping mechanism, the number of receptions should also decrease during packet routing. By doing so, we contribute in more preserving the node's resources, which helps in extending the network lifetime. For that reason, an improved sleeping mechanism has been proposed. Those will also cover proposed solutions for some problems that appeared during the deep investigation into the sleeping mechanism behavior.

- Improved sleeping mechanism: Represents an improvement of the initial sleeping mechanism. The goal of this improvement is to reduce drastically the number of packet forwarded and receptions along a routing path. Those will achieve by using several algorithms and parameters. Detailed information will be presented in 5.

Network congestion is considered the main cause of packet loss. The traditional solution was either to reduce the number of transmission packets at the source node or increase the waiting time of packet forwarding in the node's queue. By doing so, the "network delay" problem will arise. Those will negatively affect the reliability of packet reception. Therefore, a nontraditional solution must be proposed to overcome congestion during packet routing, while taking into consideration the nodes' hardware constraints in nanonetworks.

- Reducing network congestion by splitting traffic over multiple paths: Represents a packet splitting technique for overcoming congestion areas. The value in what we have proposed lies in the ability of the packets to split its route around the congestion area, through multi-paths. By deviating packets through different paths, the algorithm contributes firstly to increasing the reliability of delivering the packets to their intended destination, and in distributing the load over several paths. Detailed information will be presented in 6

2

BACKGROUND

Nanotechnology is a broad field. With my first beginnings in this field, it was necessary to allocate enough time to read the published articles to form the necessary background to start the research process. The communication in any network is based on the modulation technique and the routing protocol used to deliver the packet. It's different in nanonetwork, neither the traditional modulation techniques nor the traditional routing protocols are suitable. Therefore, specifying the modulation technique and the routing protocol to be used is a key factor.

2.1/ TS-OOK MODULATION

Nanonetworks totally differ from the traditional networks. This is due to limited hardware capabilities. So, nanonetwork is considered a new paradigm, where it is not possible to use the same routing protocol techniques and the signal modulation technique used in the traditional networks. A modulation technique TS-OOK (Time Spread On-Off Keying) has been proposed by Jornet Jornet and Akyildiz (2014) to share the radio terahertz channel for nanodevices, which is based on femtosecond-long pulses where packets are transmitted as a sequence of pulses interleaved by a given duration, cf. Fig.2.1. "1" bits are encoded with a power pulse of duration T_p and "0" bits are encoded as silence. Because sending consecutive pulses needs unavailable hardware and power at such small sizes, consecutive bits are spaced with a duration T_s which is usually much longer than the pulses themselves. Collision occurs when the receiver is receiving a "0" bit (silence), but in the same time a "1" bit arrives, which effectively shadows the "0". TS-OOK parameters can be summarized by T_p pulse duration (spreading ratio, $\beta = T_s/T_p$) and T_s (time between symbols).

The proposed duration of a pulse is $T_p = 100\,\text{fs}$ Jornet and Akyildiz (2014). Due to hardware constraints, the interval of time T_s between two consecutive bits needs to be very large compared to pulse duration. A value of 1000 has been given as example for

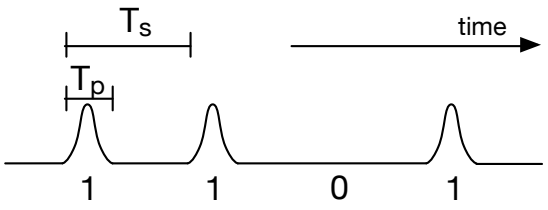


Figure 2.1: TS-OOK modulation.

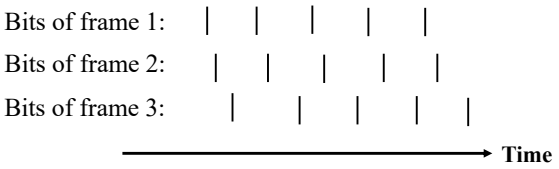


Figure 2.2: In TS-OOK, frames may overlap by bit interleaving.

the spreading ratio, defined as $\beta = T_s/T_p$ Jornet and Akyildiz (2014). Consequently, given that a bit uses an interval of time T_p, the channel accepts in theory up to $1/T_p = 10$ Tb/s, a huge bandwidth.

Besides their limited energy, nanonetworks are unique from several points of view, and some of them are important in our study. Due to their small size, numerous nodes, e.g. hundreds, can be found in the communication range of a node, leading to what we call dense networks. This could be the case for programmable matter and modular robots Thalamy et al. (2019). Also, due to the comparatively large interval of time between two consecutive bits, bits from other packets (frames) can be sent or received in between, as shown in Fig. 2.2 Dhoutaut et al. (2018). This packet overlapping or multiplexing, similar to TDMA (time-division multiple access) but at frame level allows for an extremely high channel capacity. But especially in multi-hops forwarding, it can lead to a highly inefficient avalanche of frame retransmissions if specific measures are not taken. Also, hardware constraints are of big importance. For example, those constrains prevent a node from concurrently processing more than a given number of frames, leading to ignored packets Arrabal et al. (2019) and ultimately in a huge decrease in the usable channel capacity. Finally, collisions are very specific: they do not occur on all the receivers on the channel, but only at those that receive the bits at the exact same time, which depends on their distance to source and bit propagation time. Not all collisions are destructive, for example two bits 0 do not disturb each other, given that both of them are silences Dhoutaut et al. (2018).

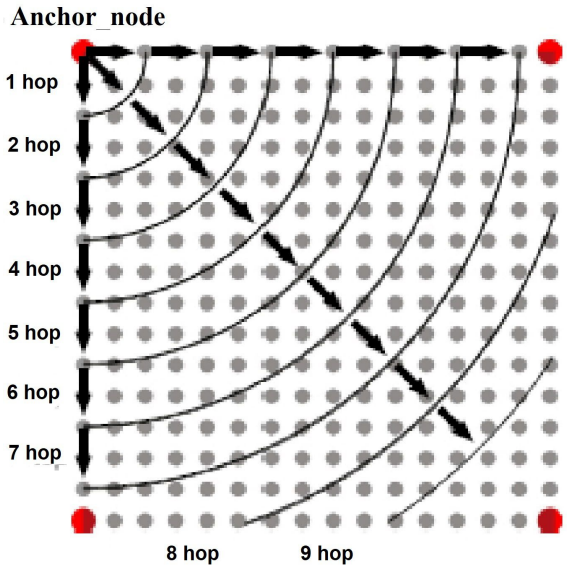


Figure 2.3: SLR addressing phase.

2.2/ SLR (STATELESS LINEAR ROUTING)

To ensure that packets reach their destination, a routing protocol is required. Nanonetworks impose more constraints to it, and the routing protocol must take into consideration the nanoscale communication's characteristics. Traditional routing protocols are not adequate for wireless nanonetworks. Differences are in terms of bandwidth, energy, and node processing capability. Designing a routing protocol becomes a challenge in WNNs (Wireless Nanonetworks) due to resource constraints on data processing, memory, and energy.

Stateless Linear-path Routing (SLR) is a spatial routing/addressing protocol. It implements a coordinate-based routing, in which data packets are routed in a linear routing path. Coordinates are defined as an integer number of hops from special nodes called anchors (Fig. 2.3). All nanonodes are assumed to be placed in a cubic space, where anchors' nodes are placed at the vertexes (two anchors for a 2D area / three anchors for 3D area).

The SLR protocol has two phases: (1) addressing/initial phase and (2) routing phase.

In the initial phase, addressing, as shown in Fig. 2.3, two anchor nodes (placed at the vertexes of 2D network) broadcast a packet (beacon) to the whole network. Beacon messages include a hop distance field initialized to 0, which increments with each retransmission. The node coordinates are the hop counter in those beacons and correspond to the distance to the anchors. This phase will be performed only once at the network deployment.

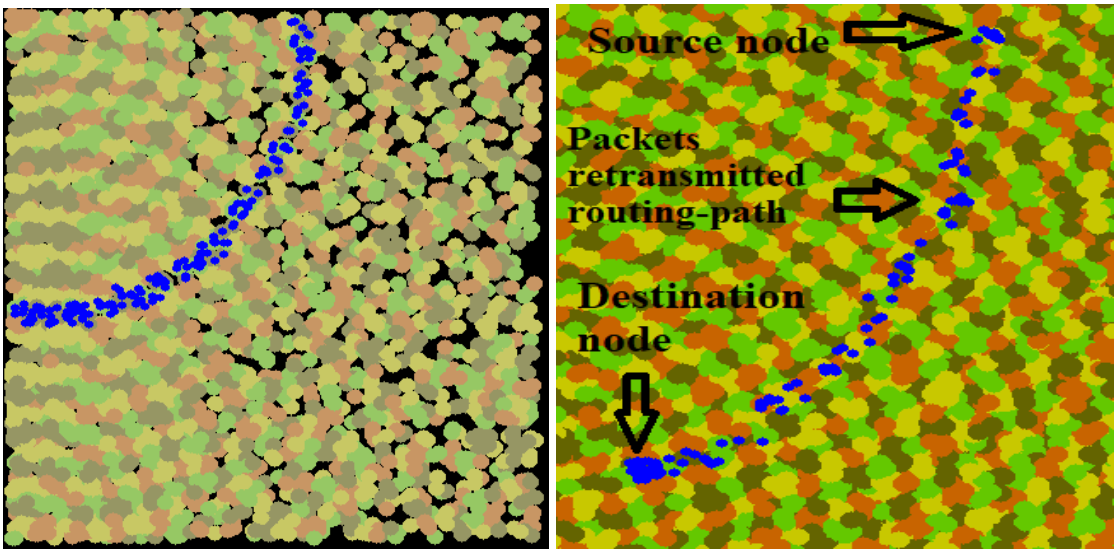


Figure 2.4: SLR initial phase (left) and SLR routing phase (right).

In the routing phase (Fig. 2.4), packets contain the SLR address of the sender and receiver. A greedy approach is used to route the packets to their intended destination. When a node receives a message, it checks if it is located on the path from the source to the destination. If and only if it is the case, the node forwards the message. This check uses basic integer computation appropriate for the low nanonode computation capabilities.

Note that because of the way the coordinates are defined, they can be shared by multiple nodes (which belongs to the same SLR zone). Consequently, SLR is a protocol that routes a packet to the correct zone instead of a specific node.

By specifying the core of any computing communication system (modulation technique and routing protocol), I am now ready to start creating and simulating scenarios. During my research period, the scenarios parameters may be changed except the modulation technique and the routing protocol kept identical in all scenarios.

2.3/ DEDEN, A NODE DENSITY ESTIMATOR

Because nodes do not have much memory or processing power, complex routing protocols that try to find optimum forwarder(s) cannot work well if at all. Hopefully, we can improve routing protocols by using the local density of the network to limit their retransmission rate. We need a way for the nodes to discover by themselves how many neighbours they have.

A traditional solution to this problem is that each node sends ”HELLO” packets. All nodes would only need to maintain a list of the neighbours from which they have received an

HELLO packet to know the local density. This solution is simple and efficient if the density is low, but it becomes prohibitively costly in resources consumption as the density increases.

Density Estimator for Dense Networks (DEDeN) Arrabal et al. (2018a) is an algorithm proposed by Femto-ST nanonetwork team, tailored to wireless nano-networks. It allows a node to estimate the number of its neighbours (also called node degree, node density, neighbour density, neighbourhood density, local density, or simply density).

DEDeN works by making nodes transmit an initialization message, and all nodes that receive it start the discovery process. With a very low (but known in advance) probability, they may start sending a few beacons. As the probability is known, nodes receiving those beacons can easily compute an estimated local density. If the confidence in the estimation in not high enough, nodes start another round, with a higher beaconing probability. The confidence increases with the number of packets received and the probability to transmit.

The confidence and the error range of the estimation can be adjusted to the requirements of the user with a predictable overhead. Depending on how it is initiated, DEDeN enables a unique node or all nodes to estimate the neighbourhood density. The algorithm may be executed each time this neighbourhood density is needed.

For additional information about the density estimator algorithm, refer to Arrabal et al. (2018a).

3

BitSimulator - programming and simulation environement

3.1/ Introduction

Researchers often use in-house simulation software developed as a side project. These tools often have flaws because of the small amount of time invested. In particular, it is quite difficult to find the correct balance between complexity, accuracy and requirements (CPU and memory). This has implications on simulation correctness. For our purposes, a simulator needs to have the following primary features.

Individual node and application instantiation. One needs a tool to help design, simulate and validate network protocols and applications. Analytical work considering the network as a whole is often not practicable (when network stack or scenario is too complex) or not sufficiently detailed (to capture subtleties and special cases). An implementation of the whole network stack is usually needed, with an individual instantiation of each element. Each node and each piece of code running is treated separately.

Bit by bit transmission and error computation. Mechanisms affecting the bit error rate but also the distribution of errors heavily depend on the coding and the payload itself. Errors need to be correctly simulated, especially when working on coding schemes.

Radio propagation delay consideration. Small changes in the position or timing in the simulated nodes significantly affect bits effectively received and collisions. Channel access control protocols that use specific binary frame preambles and compute optimal inter-bits spacing. Those protocols significantly reduce the risk of collisions but cannot rule them out, especially in very high density scenarios. Correctly simulating frames individual bits, and the timing and scheduling of events (including the propagation delay) cannot be neglected at this scale.

Frames multiplexing over the channel. This is a defining feature of wireless nanocommunications, where numerous frames (possibly hundreds or more) can be interleaved in

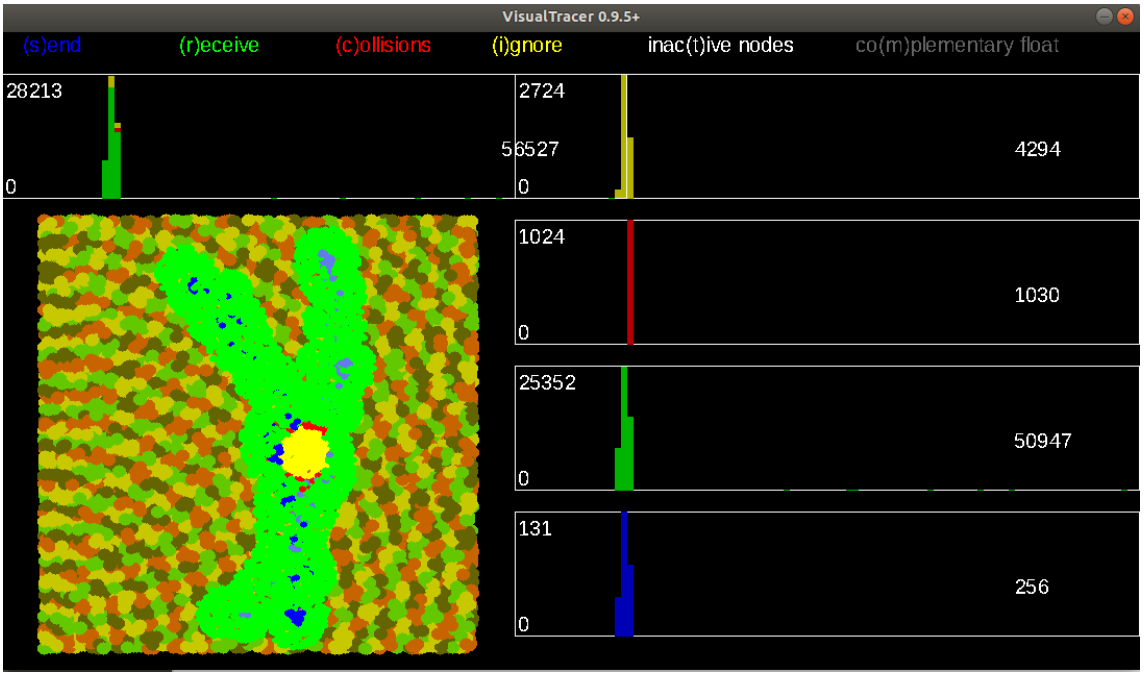


Figure 3.1: Sketch for a nanonetwork scenario (BitSimulator).

the air. This implies the ability for nodes to decode multiple frames in parallel. This is technically possible, but the number of frames concurrently decoded has to be limited to take into account hardware or software constraints.

Scaling to large number of nodes. Applications of wireless nanocommunications include programmable matter and sensor networks, which can have numerous nodes (e.g. millions). The simulator needs to be scalable with respect to the number of nodes.

Simulation speed. To avoid particular cases and unpredictable variations, a scenario is often run multiple times with different random number generator (RNG) seeds, which greatly increases simulation time. An optimized software is therefore desirable. It should be noted that fine-grained parallelization is often useless due to the interdependence of the simulated events.

Based on the above listed features, BitSimulator has been designed to allow simulation of application or routing protocols while keeping a relatively detailed model for the MAC and physical levels Fig.3.1. As such, it enables exploration and understanding of the effects of low level coding and channel access contention. For its medium access control, BitSimilator uses the TS-OOK modulation proposed by Josep Jornet Jornet and Akyildiz (2014).

3.2/ SIMULATORS COMPARISON

Several simulation programs for nanonetworks exist. However, the question remains, why do we choose this program over others. In this section, we will present the existing simulators and the characteristics of each of them.

3.2.1/ NANO-SIM

Network simulator (NS) has been in use for about thirty years. NS-2 and NS-3 are widely used in the network community.

Nano-Sim has been developed on top of ns-3 Piro et al. (2013), an emerging discrete-event and open source network simulator, designed to replace the popular ns-2 simulator in both research and educational fields. Its code is freely available under the GPLv2 license to boost its diffusion in the research community.

Nano-Sim supports the communication among nanomachines, as well as their interaction with the external world (e.g. Macroscale network, Internet, etc.). It is composed of three kinds of nodes:

- Nanonodes: small and simple devices with very limited energy, computational, and storage capabilities. They can be diffused into a target area for capturing and providing information about the environment.
- Nanorouters: nano devices having sizes and resources larger than previous ones. They are in charge of aggregating and processing information coming from nanonodes as well as controlling their behavior by means of short control messages.
- Nanointerfaces: most complex nodes able to act as gateways between the nano and the micro scale world, that is they connect the WNSN to the rest of the world. They should be able to convert WNSNs messages to a conventional network system (i.e., Wi-Fi, cellular networks, and so on) and vice-versa.

A Nano-Sim device, which is identified by a unique Dev-ID, has been conceived as a container of several entities, such as the message processing unit, the network layer, the Media Access Control (MAC) entity, and the PHY interface. The task of generating and processing messages is delegated to the Message Process Unit, which can be finely tuned based on the application requirements. Once a new message is created, the Message Process Unit sends it through the protocol stack to the physical (PHY) interface. While, when a new packet is correctly received from the channel, it will be delivered to the network layer that will verify if the message has been sent to the node. If so, the

application data will be delivered to the Message Process Unit for processing. Otherwise, the network layer could decide, according to the routing algorithm, to forward the packet.

Of course, every software has its own constraints. Therefore, after extensive research on Nano-Sim Dedu et al. (2014), we can mention some of its limitations:

- The propagation delay: this is not taken into account, when a packet is sent, all nodes in the range receive it at the same time.
- Collision: bits arriving at the same time on a receiver always cause a collision regardless of their value (“0” or “1”).
- Packets are reordered for no apparent reason, and jitter never varies, which is not realistic.
- Collision detection: it is an “all-or-nothing” calculation, i.e. if a bit of a packet collides with a bit of another packet, all bits of both packets are considered to be colliding.

Note that ns-3 implies an overhead in terms of CPU and memory, which is not negligible, drastically reducing the number of nodes that can be simulated.

3.2.2/ TERASIM

TeraSim is an ns-3 extension to simulate terahertz band communication networks Hossain et al. (2018), and released in 2018. The structure of TeraSim is divided in the following way:

- TeraSim implements one common channel module for the upper layer protocols both for nanoscale and macroscale application scenarios.
- TeraSim implements different physical layers tailored to nano and macro scenarios, namely, pulse-based communications for the nanoscale scenario and continuous-wave for the macroscale scenario.
- At the link layer, TeraSim implements two well-known MAC protocols, namely, ALOHA and CSMA, both tailored to leverage the peculiarities of the different physical layers.
- Finally, TeraSim implements some assisting modules that capture the device peculiarities, including the directional antenna module and energy harvesting module to be used mainly for macroscale and nanoscale scenarios respectively. However, these modules can be used for either scenario with minor adaptation at the MAC layer.

However, BitSimulator has several advantages over TeraSim. First, TeraSim assumes that all communications are made with the same parameter *betta*, which does not allow in-depth studies concerning this important parameter. Moreover, TeraSim does not take into account the payload of the packets, as soon as two packets overlap, if only for one bit, there is a collision and complete packets loss. This way of doing things accelerates the calculations but does not allow any study on the coding or on the video qualities for example. Finally, relying on the heavy infrastructure imposed by ns-3, TeraSim is not able to manage networks comprising a large number of nodes, a thousand nodes seems to be an optimistic limit.

3.2.3/ VOUIVRE

Vouivre is a C++ THz nano-wireless simulation library developed as both an extension for Dynamic Physical Rendering Simulator (DPRSim) and as a standalone discrete event simulator. DPRSim has been developed in the scope of the Claytronics project for supporting simulations with a large number (up to millions) of Claytronics micro-robots (also known as catoms). Original catoms do not have wireless transmission capabilities, as they are envisioned to communicate only through physical contact.

Vouivre introduces wireless communication capability to the catoms. In particular, it can be used for simulating the THz radio channel and its concurrent accesses by catoms. The THz radio channel is modeled by a continuous distance-dependent attenuation contribution increased by a certain noise value caused by the concurrent transmissions, with the noise value taken from Jornet and Akyildiz (2011). In addition, transmission delay in combination with total attenuation have been utilized for determining packet reception probability.

Moreover, Vouivre implements a TS-OOK-based physical layer, while the upper layers have not been implemented, apart from the standard CSMA/CA scheme combined with the Friss propagation model in 2.4 GHz frequency for “allowing ulterior studies of hybrid systems”.

3.3/ BITSIMULATOR

3.3.1/ INTRODUCTION

BitSimulator is a fast wireless nanonetwork bit-level simulator for routing and transport levels. It comes with a companion program, VisualTracer, allowing to show simulation results on a 2D or 3D map. Nanonetwork’s nodes are of micrometric dimensions. As such, they have drastic constraints on memory, energy, and CPU. Those constraints along

with the possibility of high neighborhood densities call for specific network protocols.

BitSimulator has been designed to allow simulation of application or routing protocols while keeping a relatively detailed model for the MAC and physical levels. As such, it enables exploration and understanding of the effects of low-level coding and channel access contention. For its Medium Access Control, BitSimulator uses the TS-OOK modulation proposed by Josep Jornet.

BitSimulator differs from other network simulators as it is completely dedicated to wireless and potentially very dense nanonetworks. Due to fine memory management and clever optimizations, simulations of up to hundreds of thousands of nodes are possible on a laptop. Of course, running time greatly varies with the complexity of the simulated scenario and with neighborhood density.

3.3.2/ FEATURES

A robust infrastructure must be built to keep the simulator simple and fast while allowing researchers to control application and network protocols. BitSimulator has many features that make it a simulation program that can compete with its peers. We can mention some of these features:

- Nano-sleep: nodes can sleep (duty cycling) with a cycle equal to T_s.
- Propagation model: two propagation models are provided: the well-known disc model, and shadowing, where the border is blurred, i.e. random losses can occur at the border of the communication range. The communication range used by nodes can change during runtime.
- Simple network scenario construction file: use of XML configuration files along with command line parameters allows easy batch runs. Changing parameters and seeds allow us to get statistical results and explore the effect of various parameters.
- Network types: can simulate in 2 different network types. The homogeneous network and heterogeneous network.
- Propagation delay: packet arrival time on a node depends on its distance from the sender (extremely important considering the duration of TS-OOK pulses).
- Collisions: computation of collisions uses the TS-OOK model by checking the actual bit value of each packet currently being received at each node.
- Focused on nanocommunications: its design is kept simple and efficient. It allows it to scale up to hundreds of thousands of simulated nodes.

- Simplicity: a simple infrastructure helps to build new routing protocols of applications (C++ classes to derive from).
- Implements several protocols: SLR routing protocol, probabilistic flooding, backoff flooding, SLR backoff flooding, Ring, and others.
- Visualisation tool: it comes with a visualisation tool (VisualTracer). At any given time, VisualTracer shows what happens in the network during the duration of the step.

3.3.3/ BECOME FAMILIAR WITH BITSIMULATOR

In order to be able to simulate any particular idea, we have to embed (by implementation) this idea inside the BitSimulator program. Therefore, it was necessary to master the work in the simulator program in order to be able to implement and test any idea that I could propose. This phase took some times to have the good knowledge about the programming structure of the BitSimulator. It was not easy because it was the first time that I tested how to deal with this kind of program, this constituted a greater motivation to discover the capabilities of the simulation program.

First of all, I got acquainted with the algorithmic architecture on which this program is built. This gave me the opportunity to get acquainted with all the programming functions within the program. Starting from the method of distributing the nodes within the network, through how to give the coordinates for each node, as well as the method used to route the packets used.

After that, I became more familiar with the way the simulation program works and started with the first implementation processes. During my work I was able to implement and simulate all the ideas and draw conclusions to be based on while writing any article.

3.3.4/ CREATE A SCENARIO AND RUN THE SIMULATION

Researchers often seek to use a simulation program that is easy and less complex, and this is beneficial to the speed of simulation and the extraction of fast and accurate results. To create a network scenario, simply you can open a file, write some parameters, including the type of routing protocol to be used, and save it in XML format.

Fig. 3.2 depicts an example of a network scenario. This scenario is composed of a network of 6mm x 6mm in size, with 20 000 nodes (medium dense). Nodes ID 0 and 1 are reserved to be used as anchors in the initialization phase of the SLR protocol. Nodes ID 2 and 3 represent a predefined source node (ID 2) and a destination node (ID 3) to

```
<scenario>
  <world sizeX_nm="6000000" sizeY_nm="0" sizeZ_nm="6000000">
    <!-- SLR anchor nodes. -->
    <node id="0" posX_nm="0" posY_nm="0" posZ_nm="0"/>
    <node id="1" posX_nm="0" posY_nm="0" posZ_nm="6000000"/>

    <node id="2" posX_nm="4500000" posY_nm="0" posZ_nm="1000000"/>
    <node id="3" posX_nm="2000000" posY_nm="0" posZ_nm="5000000"/>

            <!-- Congestion zone nodes positions -->
    <node id="4" posX_nm="3657464" posY_nm="0" posZ_nm="3477591"/>
    <node id="5" posX_nm="3794433" posY_nm="0" posZ_nm="3242820"/>
    <node id="6" posX_nm="3628741" posY_nm="0" posZ_nm="3314327"/>
    <node id="7" posX_nm="3551403" posY_nm="0" posZ_nm="3208382"/>
    <node id="8" posX_nm="3534547" posY_nm="0" posZ_nm="3257783"/>
    <node id="9" posX_nm="3762175" posY_nm="0" posZ_nm="3169769"/>
    <node id="10" posX_nm="3638495" posY_nm="0" posZ_nm="3267300"/>
    <node id="11" posX_nm="3593274" posY_nm="0" posZ_nm="3258362"/>

    <genericNodes count="20000" positionRNGSeed="1"/>
  </world>

  <modulation>
    <ts-ook pulseDuration_fs="100" defaultBeta="1000"
            defaultCommRange_nm="350000"
            maxConcurrentReceptions="10"
            minIntervalBetweenSends="0"
            minIntervalBetweenReceiveAndSend="0"/>
  </modulation>

  <routing defaultBackoffWindow="10000" backoffRNGSeed="1">
    <SLRRouting commRangeSetup_nm="220000" anchorlid="0" anchor2id="1" useDeviation="true" useCounterBasedForwarding="true"/>
  </routing>

  <applications>
```

Figure 3.2: Example of a network scenario.

determine the flow path. As for the other nodes' ID, they represent a predefined set of nodes to create a congestion area.

Recall that nanonetworks are using the TS-OOK as a modulation. A specific section in the XML file, under the name of modulation, is used to specify the modulation parameters (pulse duration, β, default communication range, etc). In addition, a part is dedicated to specifying the routing protocol that shall be used.

Fig. 3.3 shows the last part of the XML file, where the flows are created. Those lines contain the flow direction from the source node to the destination node, where the packets shall be routed.

3.3.5/ VISUALISATION TOOL

BitSimulator comes with a companion program, VisualTracer, allowing to show simulation results on a 2D or 3D map. BitSimulator provides some information (like simulation speed, simulated time, total number of events...) on the standard output. Simulation results are saved in various files. The number of packets sent, received, collided of each node is for example stored in a file under the name of results-events.log. Another file contains the position (coordinates) of each node. The analysis of the trace files produced by BitSimulator must keep it as simple as possible. Some traces are simple enough to be directly imported into a spreadsheet or into the gnuplot tool. But it is better to use a tool dedicated to visualization for larger traces. Visualtracer imports the log files produced by the simulator and produces a visualization of it. This allows an intuitive apprehension of the overall behavior of a protocol and the propagation of messages in the network, and thus facilitates the identification of possible problems in its design.

```
  <applications>
    <cbr flowId="42" srcId="2" dstId="3" port="3001" packetSize="100" interval_ns="100000" repetitions="1" startTime_ns="1000000"
beta="1000"/>
    <cbr flowId="43" srcId="24" dstId="25" port="3001" packetSize="100" interval_ns="100000" repetitions="1" startTime_ns="1000000"
beta="1000"/>
    <cbr flowId="0" srcId="5" dstId="4" port="3000" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>
    <cbr flowId="1" srcId="6" dstId="4" port="3001" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>
    <cbr flowId="2" srcId="7" dstId="4" port="3002" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>
    <cbr flowId="3" srcId="8" dstId="4" port="3003" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>
    <cbr flowId="4" srcId="9" dstId="4" port="3004" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>
    <cbr flowId="5" srcId="10" dstId="4" port="3005" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>
    <cbr flowId="6" srcId="11" dstId="4" port="3006" packetSize="1000" interval_ns="0" repetitions="1" startTime_ns="1000050" beta="100000"/>

  </applications>

  <log/>
</scenario>
```

Figure 3.3: Example of a network scenario - the flow specification.

Recall that BitSimulator comes with visualization software to graphically show what is happening during the simulation. It's a kind of transforming the results saved inside the log files into a graphical sketch. Fig. 3.4 depicts an example of a visualtracer output of a dense network scenario. Visualtracer gives clarity to the results in terms of the possibility of navigating between events. For example, we can go back or forward between events (in a time range), which gives us the ability to deeply analyze the simulation results.

The image 3.4 is characterized by a lot of colors. Each color has its own meaning. We can summarize the meaning of colors in the left zone as follows:

- Blue: represents the nodes which have just transmitted a packet.
- Green: represents the nodes which have just correctly received a packet.
- Red: represents the nodes which have just received a packet in collision.
- Yellow: represents the nodes which receive a packet, that cannot be treated.

It will be noted that during the same time interval (especially if it is large) the same node may want to display several events of different types (for example nodes receiving two packets, one received correctly and one received in a collision). To manage this kind of situation, it is possible to display/hide certain types of events. This will take effect by using the first letter of each event type to constitute the key to display or hide the state of the node. For example, if we need to display or hide the send packet event type, we can simply press on the letter ”S”.

By going to the right of Fig. 3.4, the type of the event that occurred is displayed in numbers. Each slot represents an event, for example, in this figure the number of the events that occurred at this time are: 116 sent packets (Blue), 23903 receive packets (Green), 934 collided packets (Red), and 1378 ignored packets (Yellow).

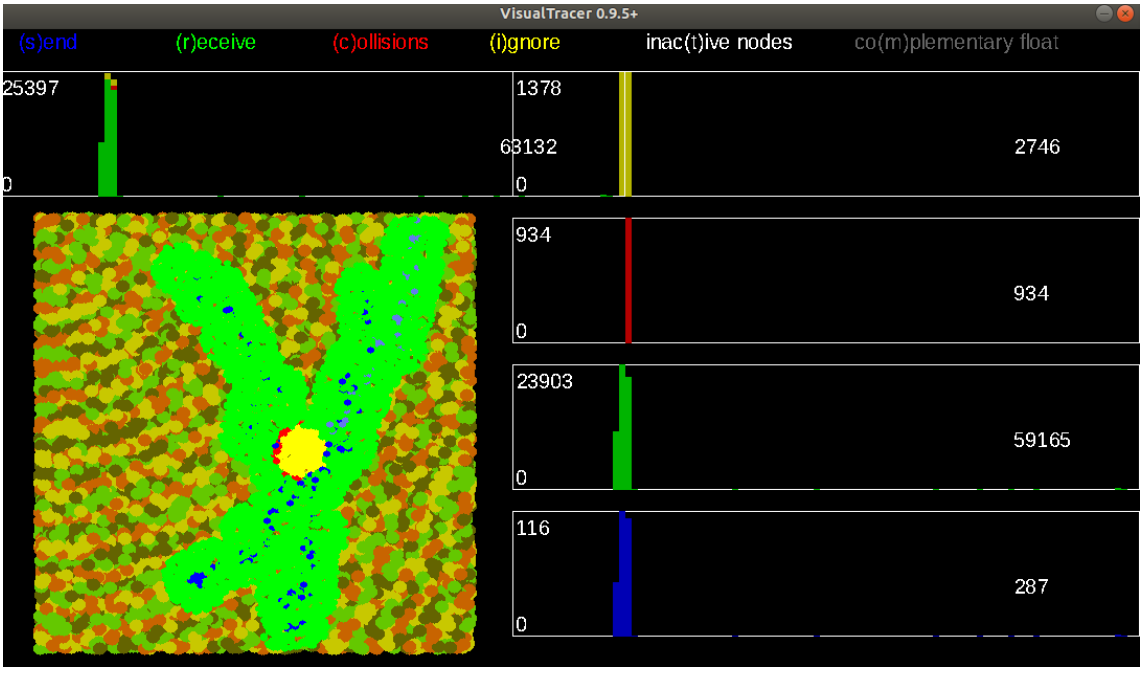


Figure 3.4: VisualTracer example of a dense network scenario.

II

CONTRIBUTION

4

Node duty-cycling (sleeping) to improve packet routing

In any network, routing and congestion control protocols play a key role, as they allow packets to reach the intended destination. We recall that a nanonode is defined as a device within the nanonetwork, that has a nanometric size (the number of nodes can be potentially dense in terms of neighboring nodes). Those nodes have limited hardware capabilities, in terms of CPU, memory, and energy consumption.

Because of its hardware limitation, a single node is not able to process the whole channel capacity. In typical networks, a node may always listen to the channel and process whatever packet concerns it. Contrarily to this, in nanonetworks there is no way for a tiny, energy constrained nanonode to sustain Tb/s traffic. This translates into a much smaller effective channel capacity, as all nodes in an area may saturate their reception capability (buffer) way before saturating the channel itself.

Nanonodes may have numerous neighbors, and due to a peculiar modulation scheme, packets can be transmitted concurrently over the channel. With many nodes and very short communications ranges, multi-hop routing and distributed protocols are required. However, their limited capabilities make it hard for the nodes to handle and route numerous concurrent packets. To cope with this, an intuitive mechanism is to distribute the load among nodes in a vicinity using a fine-grained duty-cycling (sleeping) scheme, which allows not only distributed packet processing but also distributed packet receptions, so that nodes detect and process only a fraction of the packets in the channel.

Homogeneous networks are simple and convenient to study. But with the advent of the IoT technology, and the possibility of using it in several fields, the door opened to heterogeneous networks, composed of regions of different nodes densities.

The objective of this chapter is to present a sleeping (duty-cycle) mechanism for nodes, which aims to reduce node resource usage and thereby increase the network lifetime and reduce the networks' congestion. We hereby propose a fine-grained duty cycling method

(sleeping mechanism), appropriate to nanonodes, which aims to improve packet reachability in dense networks. The present study reveals the usefulness of implementing the sleeping mechanism in both homogeneous and heterogeneous networks. In addition, we chose two ways to determine the nodes awaken duration. First, we tested a fixed awaken duration, appropriate to a homogeneous network. In heterogeneous network, where the zones have different densities, the dynamic awaken duration is more appropriate.

Implementing the sleeping mechanism in a network might lead to a packet loss problem at the destination zone by the intended node. This might occur if the packet arrives at the destination zone, while the intended node is asleep. So, we need a solution to this problem. We propose a specific algorithm to retransmit packets when they reach the destination zone.

However, allowing all nodes to participate at the packet retransmission process has a negative impact, and might led into create a congestion area at the destination zone. For that reason, we propose an enhanced retransmission algorithm used by the nodes in the destination zone, in combination with our previously proposed nano-sleeping mechanism. This algorithm increases the chance of a destination node capturing the intended packet while decreasing the number of participating nodes in the retransmission process.

4.1/ INTRODUCTION

Wireless nanosensor networks are built from tiny nodes, equipped with embedded computing, sensing and actuating devices. They usually have small CPU, small memory and low battery. Upcoming nanosensor networks are expected to use electromagnetic waves from the THz band (0.1–10 THz) for their communications Yao et al. (2019). Due to power constraints, they will also have to use multi-hop communications to deliver the packet to its intended destination.

In a traditional dense sensor network Zhao and Govindan (2003), congestion is the state where the number of packets exceeds the network capacity. Some of them are discarded or lost due to collisions. Contrary to traditional wireless networks, congestion in nanonetworks Kafi et al. (2014) does not arise from a saturated channel, but from an insufficient capacity of individual nodes on the multi-hop path to process all the incoming concurrent packets.

Routing protocol is responsible to route packets from source to the desired destination. As mentioned before, during packet routing and due to network congestion, packets may not reach their destination, which makes the communication in the network unreliable. Traditional routing protocols are not adequate for wireless nanonetworks. Differences are in terms of bandwidth, energy, and node processing capability. Designing a routing

protocol becomes a challenge in WNNs due to resource constraints on data processing, memory, and energy. While designing new routing protocols, the following points must be considered:

- Energy efficiency: nanonodes are battery-powered. In low dense environments, and where there is a high rate of data exchanging, energy shortage is a major issue. Therefore, the routing protocol should be energy efficient Yin et al. (2008).
- Node density: nanonetworks could be of different densities (low, medium, high, ultra-high, where nodes have numerous neighbours). Therefore, the routing protocol must support various network densities.
- Complexity: due to limited hardware capability and resources, the complexity of a routing protocol may affect the performance of the entire WNN. The lower the complexity, the highest its effectiveness.
- Delay: in some applications, the delay, defined as the time taken to transmit the data from the source node to the destination node, is a key factor in message receiving or response. Therefore, the routing protocol should provide a reasonable delay.

SLR (Stateless Linear-path Routing) Tsioliaridou et al. (2017) is the protocol we use in all our evaluations. We recall that it implements a coordinate-based routing, in which data packets are routed in linear routing path. Nodes are assumed to be placed in a cubic space, distributed in zones. In the initial SLR phase, during network deployment, a few anchor nodes broadcast a packet (beacon) to the whole network. The hop counter in those beacons is used to define the coordinates of all nodes as a distance to the anchors. In the second phase, during data packet routing, nodes choose to forward a packet if and only if they are on the path between the source and the destination of the packet.

We also recall that nanonetworks use the TS-OOK (Time Spread On-Off Keying) modulation Jornet and Akyildiz (2014) to share the radio terahertz channel for nanodevices, which is based on femtosecond-long pulses where packets are transmitted as a sequence of pulses interleaved by a given duration. “1” bits are encoded with a power pulse of duration T_p and “0” bits are encoded as silence. Because sending consecutive pulses needs unavailable hardware and power at such small sizes, consecutive bits are spaced with a duration T_s which is usually much longer than the pulses themselves. Collision occurs when the receiver is receiving a “0” bit (silence), but in the same time a “1” bit arrives, which effectively shadows the “0”. TS-OOK parameters can be summarized by T_p (pulse duration), β (spreading ratio, $\beta = T_s/T_p$) and T_s (time between symbols).

Even if the channel capacity is very high, nodes have very low capabilities and cannot completely use it individually. Due to the TS-OOK’s ability to interleave packets over the

channel, nodes can receive multiple packets concurrently. However, the number of packets that could be received concurrently is bound to be limited, due to technical constraints. We call this limitation emanating from CPU, memory or other hardware maximum concurrent receptions. In the following evaluations, we will consider this limitation to be equal to 5. This means that while 5 packets are already being received, new incoming ones will be silently ignored. This is different from collisions, as the packets being received are not affected. This mechanism nevertheless means that nodes can very easily be saturated in a given area, effectively reducing the available network capacity and producing a new form of congestion.

Keeping a nanonode awake for the whole T_s duration makes it easy to saturate. Our idea is to make it inactive for a fraction of T_s, effectively implementing a very fast and sub-packet level duty cycle. Thus, the node will not hear all incoming packets anymore. But we compensate this by having the other nodes asynchronously apply the same mechanism. By doing so, incoming packets are handled and routed only by awake nodes.

We recall that nanonetwork(s) can be potentially used in a wide range of applications and fields such as: (1) health field (e.g. monitor human vital signs, drug delivery systems), (2) military field (e.g. border control to prevent any penetration), (3) agricultural field (e.g. monitoring the temperature, humidity, and water level of plants), (4) telecommunication sector Tsioliaridou et al. (2017). In such applications the network topology is often heterogeneous. A heterogeneous nanonetwork is a network composed of zones of different densities. For example, in agricultural applications, some areas have high nodes' density while others have a small density due to the varied geographical terrains. Also, the human body is considered as a natural heterogeneous network because of its internal and external structure. Thus, many applications use this type of networks.

4.2/ Related work

4.2.1/ Sleeping in macroscale networks

To preserve nodes' resources, a well-known solution in macroscale networks is to put the nodes to sleep. Usually, the time when the nodes will be asleep is predefined. The nodes will stay awake for a long enough time to send and receive packets, afterwards, they will be asleep based on the predefined asleep time.

With the developments of the networks, a new type appeared called WSN (Wireless Sensor Networks). Thus, are used in different areas with several applications. WSNs have limitations in terms of energy, they are powered by small batteries in most applications Yick et al. (2008). To extend the network lifetime and reduce energy consumption, sleeping strategy has been deployed in WSN, it was showing its effectiveness on pre-

serving nodes resources Nan et al. (2012).

In literature, several sleep/awake schemes have been proposed to preserve sensor nodes' resources. While applying the sleeping mechanism, a time orchestration should be applied to manage the sleep and awake of nodes. But with limitations of the sensor hardware in terms of processing and memory, this becomes a challenge. The author in this article Timmons and Scanlon (2011) proposed an algorithm that uses adaptive band guards to allow nodes to sleep through several periods without synchronization.

By switching to another type of wireless network (cellular networks) and due to the rapid growth in telecommunication applications, the users have more tendency to interact with the video call platforms, share multimedia files, etc. Therefore, providing better coverage for users and guaranteeing a good quality of service became essential. The good coverage and the high quality of service are achieved by installing more base stations (BTSs). This, lead to augments in the power consumption in the cellular network, therefore the increase in operating cost. In Chen et al. (2017), deep and profiteering methods were respectively proposed to find the optimal number of SBS (Small Cell Base Station) and antennas that should be switched off in dense heterogeneous networks. According to the deep sleep method, SBSs with zero or low load are switched off and stay in deep sleep mode during a predefined time interval. By doing so, some antennas are turned off, and the remaining antennas jointly serve users via cooperative transmission.

An optimized clustering-based sleep strategy is proposed to release the power consumption and interference in the system Li et al. (2018). The introduced algorithm (BPSO: binary particle swarm optimization) forms the clusters of SBSs based on the interference, and the sleeping method is applied to each cluster separately. The small base stations with large interference are grouped as a cluster, where the BPSO algorithm is applied to orchestrate the sleep strategy for SBSs in the clusters.

Wi-Fi as a technology has been a game-changer for expanding the internet. Today, having Wi-Fi in our homes, offices, and other public places is very common. Therefore, reducing power consumption becomes a key factor due to its widespread.

In WLANs, wireless devices such as laptops, smartphones and tablets are equipped with the Wireless Network Interface Controller (WNIC). WNIC allows wireless devices to share, communicate and access information wirelessly through an Access Point (AP) Afaqui et al. (2017). Unfortunately, one of the most important outstanding issues reaming in WLANs is the power consumption caused by WNIC during its communication with an AP. The high level of power consumption during the communication of WNIC directly affects the battery life of a wireless device, if is not connected to a power outlet Saha et al. (2015).

The 802.11 standard defined a Static Power Save Mode (SPSM). In SPSM, a WNIC of a wireless device can sleep to save energy, and wake up periodically to receive its buffered

packets from an AP Monteleoni et al. (2004). Unfortunately, SPSM highlights several issues as the frame overhead and delayed delivery of packets when the WNIC is off between the beacon intervals. Those will affect the performance of interactive applications such as real-time video and audio applications.

To enhance the lacks in SPSM, the authors in Pyles et al. (2012) propose a smart algorithm based on applications priority. Smart Adaptive PSM (SAPSM) labels each network-based application of smartphone into two set of priorities; high and low, with the help of a Machine Learning (ML) classifier. Thus, for applications set as high priority, the WNIC will be adaptively switched into awake mode, and stays in the SPSM with applications set as low priority.

4.2.2/ SLEEPING IN NANOSCALE NETWORKS

Recall that when the sensor nodes are awake and without any particular configuration, they listen to the channel continuously for the reception of new packets. These packets must be processed by the node, even if the node is not the destination of the packet. This situation has a significant impact on the energy consumption of the nodes.

Applying the sleeping mechanism is highly required in nanoscale networks. This is due to the fact that the nodes are limited in processing, memory, and energy capabilities. By doing so, we contribute in preserve nodes' resources and extending the network lifetime.

Singh and Raghavendra Singh and Raghavendra (1998) proposed an energy-efficient protocol called Power-Aware Multi-Access Protocol with Signaling (PAMAS) based on Multiple Access with Collision Avoidance (MACA) Karn et al. (1990). The PAMAS has incorporated a new signaling mechanism over MACA. The idea is to turn nodes into the sleep state when they are not transferred or relaying packets. However, the authors in Singh and Raghavendra (1998) do not consider the node's idle state listening which too consumes a significant amount of power.

Like in all shared-medium networks, medium access control (MAC) is an important technique that enables the successful operation of the network. In Ye et al. (2002) authors propose three techniques to decrease energy consumption under the name of S-MAC. First, adopting periodically sleep to preserve energy when nodes are listening to an idle channel. Second, based on PAMAS, S-MAC also sets the radio to sleep during transmissions of other nodes. Finally, the message passing technique is applied to reduce latency for sensor-network applications.

Authors in van Dam and Langendoen (2003), propose an algorithm to handle load variations in time and location. T-MAC introduces an adaptive duty cycle in a novel way: by dynamically ending the active part of it. This lead to reduce the amount of energy wasted

on idle listening, in which nodes wait for incoming messages, while still maintaining a reasonable throughput.

The above listed mechanism relies on clock synchronization, neighbor discovery, etc. Recall that nanonodes resources are restricted in terms of processing ability, energy and memory. Therefore, it was necessary to find a sleep mechanism that does not exhaust the nodes' resources by additional computing tasks, but rather helps to preserve resources and prolongs the network lifetime.

4.2.3/ FLOODING METHODS

Pure flooding In ad-hoc wireless networks, multi-hop data broadcasting is an essential service. It is required by several applications, and used to broadcast information in the network (e.g. routing table updates, path updates, etc.). Flooding is important especially in mobile ad-hoc networks (MANETs), which rely on it to perform routing discovery.

One of the traditional routing methods is pure flooding Miller et al. (2005). It is motivated due to its simplicity, which conforms to the constraint capabilities of the nanonodes. It is an unreliable operation with no acknowledgment mechanism in place. In pure flooding, a node forwards each message (without routing data) received for the first time. However, this technique has drawbacks, the most notable is the generation of a significant amount of messages in the whole network. In dense networks, propagation growth leads to a broadcast storm. Countermeasures have to be taken to prevent skyrocketing contention for channel access or collisions.

Probabilistic flooding Many attempts have been made to optimize the pure flooding technique by selecting a subset of forwarding nodes. They are challenged by nanonode hardware limitation, either by the inability to build a complete map of even the direct neighbors, or because of too high memory requirements.

A common solution is to give each node a probability to forward a new packet (already seen packets by a node are discarded anyway). The probability chosen could be fixed, or depend on several factors, such as density, distance, speed, and others. The most considered metric in calculating the probability is the number of neighboring nodes.

Probabilistic flooding greatly reduces redundant retransmissions and receptions compared to the pure flooding scheme. Several probabilistic flooding schemes have been proposed for wireless ad-hoc networks that require lightweight computing resources. Hence, they can be used for data dissemination in nanonetworks Reina et al. (2015). One of the most important defects they have is the die-out problem Arrabal et al. (2018b).

4.3/ NANO-SLEEPING MECHANISM

4.3.1/ FINE-GRAINED NANONODE DUTY-CYCLING

A common technique to preserve node resources (e.g. processing of fewer packets, memory, energy) is to use duty-cycling (sleeping) techniques, where nodes wake up from time to time to receive and process packets, and sleep otherwise.

The sleeping schemes in classical networks are coarse-grained: nodes stay awake for "large" periods of time, allowing them to receive (or send) at least one complete frame. Alternatively, we hereby propose Medlej et al. (2020) a fine-grained duty-cycling mechanism which considers nanonetwork peculiarities. This mechanism differs from those used in macro-scale networks on three main aspects:

- Not necessarily an energy-saving sleep: the point of this duty-cycling is to controllably ignore a part of the incoming packets. We do not intend to shut down the device in order to save energy. In fact we are aware that it would not be possible at such a high frequency. Here, "sleeping" can be as simple as temporarily disconnecting the antenna.
- Fine granularity: a nanonode does not stay awake for the duration of one or several packets, but for a much shorter duration, a fraction of the T_s value (the interval between two consecutive bits). This allows it to receive only one bit of a given packet at a time, and potentially a few bits that belong to other packets, due to packet interleaving. We recall that nodes are highly synchronised in TS-OOK, and we assume that the issues related to the fast awake-sleep switching are negligible.
- Asynchronism, decentralization: there is no agreement among nodes on awake periods. A node simply receives the bits that arrive when it is awake, whether they are intended for itself or not, and misses all the other bits.

In the fine-grained sleeping mechanism we propose Medlej et al. (2020), all the communications use the same spreading ratio β, and all the nodes have the same awake-sleep cycle, equal to $T_s = \beta T_p$. Inside the cycle, the beginning of the awake interval is different for each node, which chooses it randomly. The awake duration (or percentage of T_s) is either the same for all nodes, or nodes can have their own duration (we explore both cases in the coming sections). Fig. 4.1 shows an example of three receivers waking up at different times, but with the same awake duration, all of them having a cycle of T_s.

It is important to note that this mechanism ensures that packet reception by a node is all or nothing: given that the cycle length is T_s, the mechanism ensures that if a node is able to pick the first bit of a packet, it will be able to pick all the following ones. By adjusting the

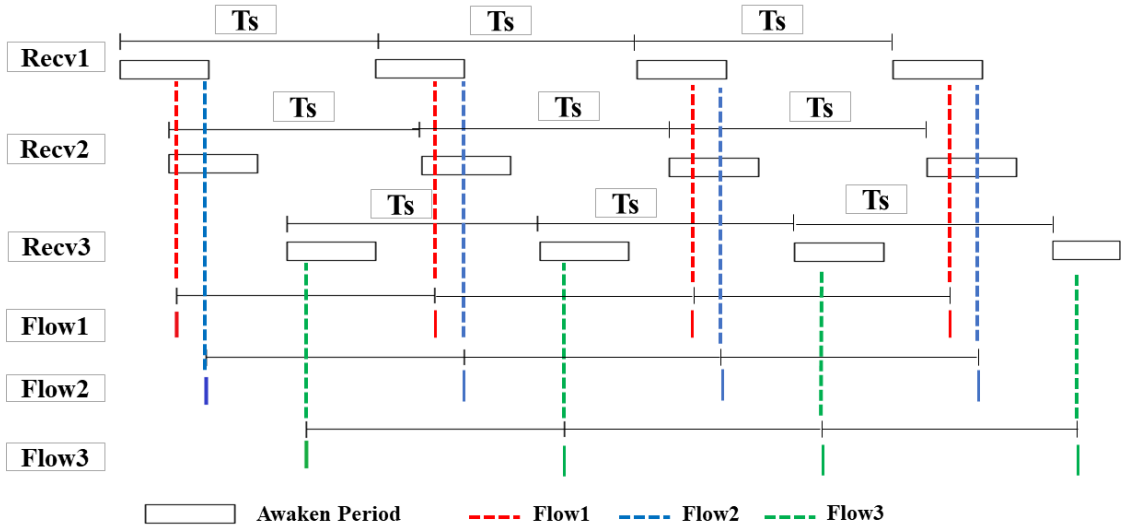


Figure 4.1: Sleeping mechanism with three nodes and three flows, with the same awaken duration.

awake duration to the local density of the network, one can also statistically assure that a packet will be received by at least one or a few nodes (without knowing which ones in advance). In Fig. 4.1, Recv1 and Recv2 see all the bits from flows 1 and 2, as they arrive when they are awake; Recv3 sees bits only from flow 3.

The sleeping mechanism can be implemented or mimicked in different ways:

- Nodes effectively sleep, consuming less energy. They wake up only at the predetermined time;
- Nodes do not sleep, but simply do not take into account the bits received in the ”sleeping” period; this process is still advantageous, because nodes process fewer frames.

We mostly target nanonodes and their extremely short pulses. The inter-pulses duration inside a frame is so short that a real energy-saving duty-cycle is not feasible. Consequently, we rather consider that the second mechanism is used, where sleeping means that nodes just stop listening for some periods of time (or discard packets arriving during that interval of time).

4.3.2/ SLEEPING AND DESTINATION NODE CONSIDERATIONS

Sleeping improves network behavior by limiting the amount of traffic an individual node can see. Traffic is statistically dispatched over all nodes, thus sharing the load. This mechanism is especially effective while routing packets. But provisions have to be made

to ensure the destination node is not sleeping when packets arrive its neighbourhood and missing some data packets.

A first way to deal with this problem is to ensure the destination node is not sleeping at all. This is the case for networks where the destination node is in a relatively calm area. As not much competing traffic comes near the destination, it does not saturate and can stay awake all the time.

In case of a destination being in a very loaded area, a better strategy is to not target a specific node, but instead ”any node providing the required service” in the *destination zone*. This way, it becomes possible to have multiple nodes implementing the service and sharing the same SLR coordinates. Those nodes will sleep asynchronously, with an awake time percentage depending on how many they are. At any time, some of them will be awake and able to pick any incoming packet.

The way is different when the goal is to deliver the packet to a specific node. There are two choices in this aspect. The first choice is where all the nodes at the destination zone will retransmit the packet (full retransmission) at the end of its awaken duration, in that way the intended node will have the chance to receive the packet even after a while. More details are presented in 4.3.2.1. In the second choice, not all nodes will participate in packet retransmission. Instead, we rely on the nodes' local density to determine the number of participating nodes in packet retransmission (probabilistic retransmission). More details are presented in 4.3.2.2.

Another consideration is that usually the best awake percentage is the one that has the lowest number of forwards, receptions, ignored packets, while maintaining a good probability of packets arrived at the destination.

However, there are applications which do not need a 100% reliability (reachability to the destination). For example, in agriculture applications, nanonodes can be used to measure plant life indicators (such as humidity, soil moisture, sun light) and transmit this information to a central server for processing. Whereas such applications need a high network life time, the reliability of arrived packets is not critical, since any lost information will be resent anyway. Our sleeping mechanism helps such applications by improving the network life time using a small awaken period.

Some critical applications rely on the high reliability of packet delivery. Therefore, special work should be applied at the destination zone to increase the chance of packet receptions.

4.3.2.1/ FULL RETRANSMISSION

The definition of destination may change depending on the application. In some applications, ensuring a packet reception by a specific node is crucial. For example, *Target Drug Delivery (TDD)* can be explained as a daily therapeutic action. The concept of target drug delivery (TDD) is employed to address the toxicity and wastage challenges. The TDD approach is geared towards ensuring that therapeutic drugs are localized to a targeted (pathological) part of the body in a controlled manner using various techniques without affecting other healthy parts of the body where they will otherwise produce adverse effects. It is also aimed at minimizing drug degradation and loss.

In TDD scenarios, drug localization to targeted cells is achieved by encapsulating, entrapping, or encapsidating drug and therapeutic molecules/particles or agents in nanovectors. For some reason, if the nanovectors do not receive or collect information, this might put the patient at risk. Therefore, having an information reception algorithm of high reliability became a key factor in IoT applications, especially in the medical field.

Another example where the packet reception reliability is a key factor is the periodic car maintenance application. Periodic car maintenance can be explained as a service/maintenance model, where a car undergoes a service/maintenance either after a certain specified time period or on the basis of a part getting faulty Dhall (2017). Car parts (brake, motor, etc.) are equipped with a sensor for monitoring and data collecting purposes. For some reason, if the brake sensor does not receive or collect information, this might put the driver at risk in case the brakes are faulty. Therefore, having a packet reception algorithm of high reliability became a key factor in IoT applications.

If the destination is defined as a node, it means that we want the packet to reach the destination zone and the intended node receives this packet. In that case, a mechanism should be proposed to achieve this goal. For that reason, we propose a full retransmission algorithm 1 that aims to retransmit the packet by each node at the end of its awaken duration, at the destination zone.

The algorithm works on homogeneous and heterogeneous networks, with equal or different awaken durations. In this algorithm we took into consideration the case when the awaken duration spans over two time cycles. We describe the algorithm as follows:

- waiting*T*ime: represents the time the node should wait before packet retransmission at the end of its awaken duration.
- w*T*1ts: formula to calculate the node waiting time if the awaken duration range is 1 T_s.
- w*T*2ts: formula to calculate the node waiting time when its awaken duration spans for 2 T_s.

Algorithm 1 Packet retransmission algorithm executed by all nodes at the destination zone except the destination node.

```
alreadyseen = false
waitingTime
wT1ts = aD - (pcktrecp - aS)mod T_s
wT2ts = - (aD - (pcktrecp - aS))mod T_s
if packet type is data and reach the destination zone then
    if Sleeping mechanism is used then
        if packet !alreadyseen and the received node is not the destination node then
            if pcktrecp mod T_s ≥ aS then
                waitingTime = wT1ts
            else pcktrecp mod T_s ¡ aS + aD - T_s
                waitingTime = wT2ts
            end if
        end if
    end if
end if
```

- aD: represents the node awaken duration.
- aS: represents the node awaken starting time.
- pcktrecp: represents the time when the node receives the packet.

Fig. 4.2 depicts an example of a packet retransmission at the destination zone. In our proposed algorithm, all the nodes have the same awake-sleep cycle T_s. Inside the cycle, all the nodes are different in their awake duration aD (or percentage of T_s), also the beginning of the awake interval for each node aS are randomly determined. When a node receives the packet, the algorithm checks whether the node is not the destination node. If this is the case, the algorithm will be applied.

Node 3 represents the destination node. Node 1 receives the packet, and since it is not the destination node, it executed the algorithm. The algorithm computes the waiting time to retransmit the packet at the end of the node awaken duration. By coincidence, nodes 2 and 4 are awake during the packet retransmission by Node 1. It happens that a node can receive several copies of the packet, which is the case for Node 4. This is due to the simultaneous awake of the node with the retransmission.

Finally, the figure shows that the destination node awaken time does not overlap with any of the other nodes. Node 4 retransmits the packet at the end of his awaken duration, and it is received by the destination Node 3. Therefore, in the absence of the retransmission algorithm, the destination node would not have received the packet in the case where it was in sleep mode.

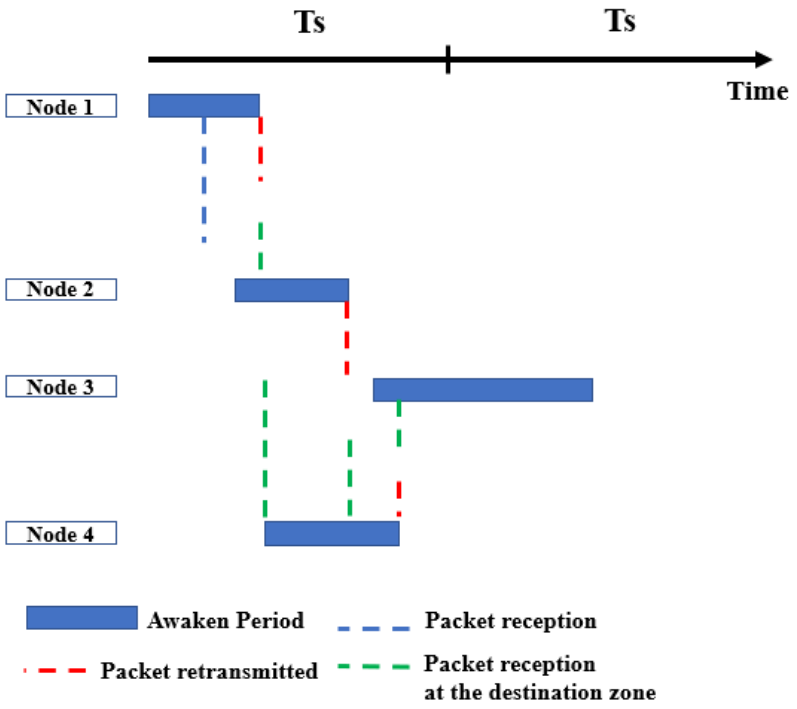


Figure 4.2: The effect of the retransmission algorithm at the destination zone.

4.3.2.2/ PROBABILISTIC RETRANSMISSION

Making all nodes at the destination zone retransmit the packet leads to nodes' resources being exhausted. To avoid this problem, we propose an enhanced packet retransmission algorithm (probabilistic retransmission), where not all the nodes participate in the retransmission mechanism. The number of participating nodes is determined based on a probability, calculated as follows:

$$probability = 1 - \frac{aD}{T_s} \tag{4.1}$$

In this formula, the retransmission probability is inversely proportional to the awaken duration percentage aD. Table 4.1 shows the expected number of participating nodes among various awaken durations. In this network, we have used 25000 nodes randomly distributed. The destination zone includes 41 nodes. The table shows that while node awaken duration increases, the number of participating nodes decreases. More details are detailed in section 4.4.3.2.

No matter the network density, the probabilistic retransmission algorithm, will never saturate the radio channel and does not require much memory, or computations. The only memory needed is the buffer to store the received packet to retransmit it at the end of the awaken duration.

In this algorithm, we took into consideration the case where the awaken duration spans

Table 4.1: The expected number of participated nodes in full and probabilistic retransmissions.

Awaken duration (%)	Full retransmission	Probabilistic retransmission
6	35	35–38
10	41	34–37
20	41	32–35
30	41	28–32
40	41	25–28
50	41	21–25
60	41	15–21
70	41	11–15
80	41	7–11
90	41	3–7
100	41	0

over two-time cycles. The variables used in this algorithm kept identical to what we used in 1, the additional parameters are the following:

- probaRNG: a probability random number generator function (0,1).
- proba: the calculated probability based on node awakenDuration.

The node retransmits the packet if and only if the probaRNG random variable is less than the calculated probability. The enhanced packet retransmission algorithm is presented in Algorithm 2.

To the best of our knowledge, there is no similar proposed algorithm that takes into consideration achieving a reliable packet reception at the destination (zone/node). The evaluation results are detailed in 4.4.3.

4.4/ EVALUATION

This section shows the effectiveness of our proposed algorithms and their positive effects on the nanonetwork.

4.4.1/ DUTY-CYCLING IN HOMOGENEOUS NETWORKS WITH DESTINATION ZONE

In a homogeneous network, the nodes are randomly distributed within the network. In this section we show the effectiveness of applying the sleeping mechanism in reducing the number of forwarders inside the network.

Algorithm 2 Probabilistic retransmission algorithm executed by nodes at the destination zone only.

```
alreadyseen = false
waitingTime
wT1ts = (aD - (pcktrecp - aS)) % T_s
wT2ts = - (aD - (pcktrecp - aS)) % T_s
if packet type is data then
    if packet !alreadyseen AND the received node is not the destination node then
        alreadyseen = true
        if pcktrecp % T_s ≥ aS then
            waitingTime = wT1ts
        else // pcktrecp % T_s < aS + aD - T_s
            waitingTime = wT2ts
        end if
        probaRNG = rand (0, 1)
        proba = 1 – (aD / T_s)
        if probaRNG < proba then
            the node will retransmit the packet at the end of its aD (now + waitingTime)
        end if
    end if
end if
```

Table 4.2: Simulation parameters.

Size of simulated area	6 mm * 6 mm
Number of nodes	3000 to 15 000
Communication radius	350 µm
β (TS-OOK time spreading ratio)	1000
T_p	100 fs
Packet size	1000 bit

4.4.1.1/ NETWORK SCENARIO

The simulation parameters are shown in Table 4.2. The network is a 2D area. The main flow has the source at the bottom of the network, and the destination at the top. The source sends 100 unique packets to the destination. Several interfering flows (more precisely, 92, with specific parameters β = 1000, packet size = 1000 bits, interval time between packets is 0), if any, cross the network from left to right of the network, as shown in Fig. 4.3.

The communication range is chosen so that several hops are necessary to cross the network. Indeed, the mono-hop communication surface is a disc $\pi 0.35^2 \approx 0.38\,\mathrm{mm}^2$, so the simulated area contains $6*6/0.38 = 94$ disjoint discs. Routing is done through the SLR protocol 2.2. Fig. 4.3 shows a map where each colored zone corresponds to a different distance expressed in hop count from the SLR anchors. The node density is high, i.e. from $3000/94 \approx 32$ to $15000/94 \approx 160$ neighbors.

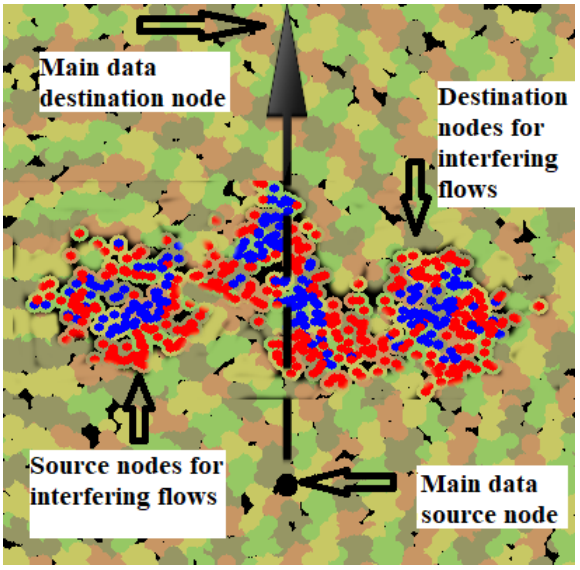


Figure 4.3: VisualTracer output for the evaluated network.

In the following analysis, all the nodes use the sleeping mechanism. We vary the number of randomly positioned nodes and the number of competing data flows that may interfere with the main flow. To avoid random effects, each point in the following figures is the average of 15 simulations using different RNG seeds, used for the position of all the nodes except source and destination ones, all the other parameters being kept identical.

4.4.1.2/ COMPARISON METRICS

The sleeping mechanism uses less node resources (CPU, memory, energy) but can prevent packets to arrive to the destination. Therefore, the metrics we use are the following:

- Arrived packets: the number of packets that have reached the destination node; only unique packets from the 100 original ones reaching the destination are counted, otherwise said multiples copies of the same packet are counted as one.
- Main data flow forwards and receptions: the number of times the packets of the main data flow are sent or received by a node in the network (at ”MAC” level).
- Forwards and receptions: the total number of forwards and receptions, i.e. for the main flow plus the interfering flows; this metric is a good indication on the energy used.
- Collided and ignored packets: a collision occurs when packets arrive at the same time and hence prevent correct decoding by the receiving node. Ignored packets are discarded due to insufficient capacity to process all the incoming concurrent

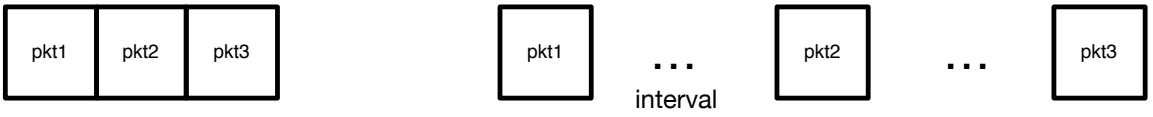


Figure 4.4: Packets sent without interruption or sent at a fixed interval.

packets (buffer overflow or other physical limitations). Nodes are aware of collisions, but may be unaware of ignored packets.

4.4.1.3/ RESULTS ANALYSIS: WITHOUT INTERFERING FLOWS

In SLR, nodes receiving a packet check if they are on the path from its source to its destination, and forward the packet if this is the case. But multiple nodes may share the same coordinates in SLR and they will all take the same decision to forward. In very dense networks, as many neighbouring nodes try to forward at the same time, this can cause collisions and ignored packets.

In this scenario, the source sends 100 packets (1) as fast as possible (one after the other), or (2) with an interval between packets equal to 5 times the duration of a packet (see Fig. 4.4). In such a multi-hop scenario, sending as fast as possible is quite demanding, as copies forwarded by neighbours tend to interfere with new incoming packets. Slowing the transmission speed even if data are available at the source is a good strategy, especially as nodes are not expected to send long bursts of data.

When inter-packet waiting period is null (Fig. 4.5), only around 40% of unique packets reach the destination for always awake nodes (100% on horizontal axis). The more the nodes sleep, the higher the reachability, up to the point where not enough nodes are awake and the transmission becomes impossible. Finally, denser networks benefit even more from the sleeping mechanism.

In case of an interval between packets, the intensity of the data burst increases, and the reachability improves greatly. In Fig. 4.6, at 100% of time awake, all the packets are received. Also, decreasing the awake time does not immediately reduce the number of arrived packets, giving space for improving the network usage. This is especially true for very dense networks. For example in Fig. 4.6, in the densest network (15 000 nodes), 10% of awake time is sufficient to allow all the packets to arrive at the destination.

Fig. 4.7 and 4.8 highlight an unexpected behavior of our sleep mechanism. As expected, there is a collapse of forwards and receptions when the awake time is very low and the data packets cannot be forwarded anymore. But for larger percentages of awake time, curves are mostly stable. This is contrary to what Fig. 4.6 showed, i.e. the reachability improves with the decreasing of the awake time.

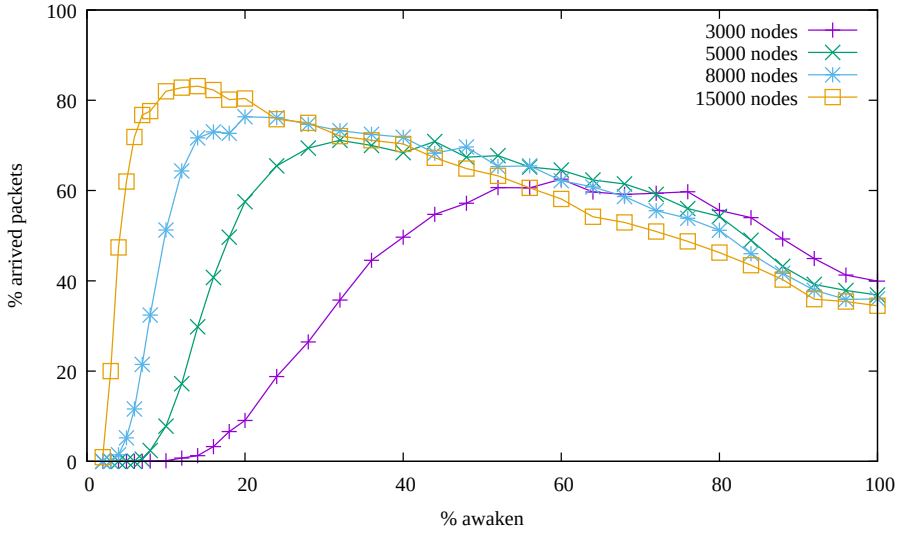


Figure 4.5: Percentage of arrived packets depending on awake time and network density, for one flow and no interval between packets.

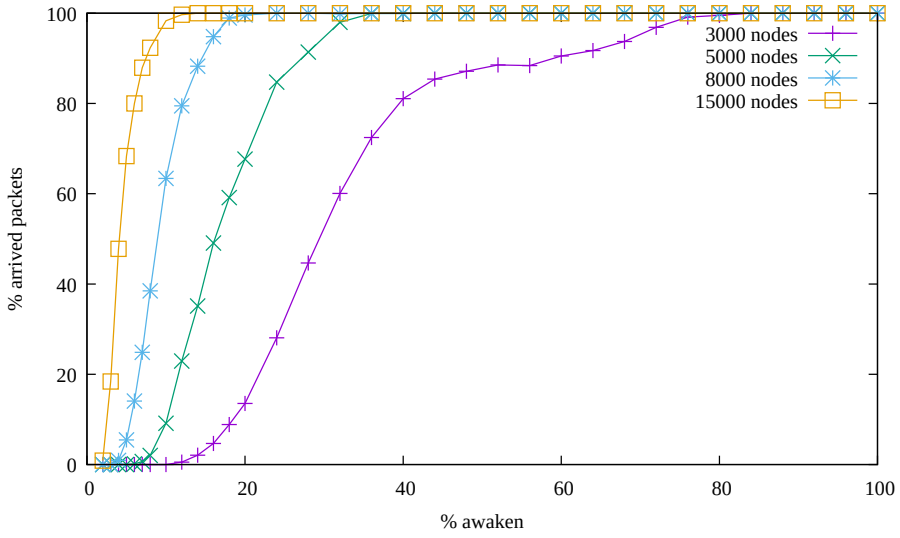


Figure 4.6: Percentage of arrived packets depending on awake time and network density, for one flow and with interval between packets.

A stable or even an increase in transmission has a twofold explanation: First and foremost, the sleep mechanism dispatches the load of forwarding packets among neighboring nodes. As not all of them receive at the same time, they are not saturated (ignoring packets) at the same time. The same number of nodes becomes able to forward more data packets. This effectively increases the capacity of the network. This is an important result of this evaluation. As a second reason, we can also note that as packets are not lost (cf. Fig. 4.5 and 4.6), they can be repeated along the whole path. This effectively increases the number of times they are counted as forwarded packets.

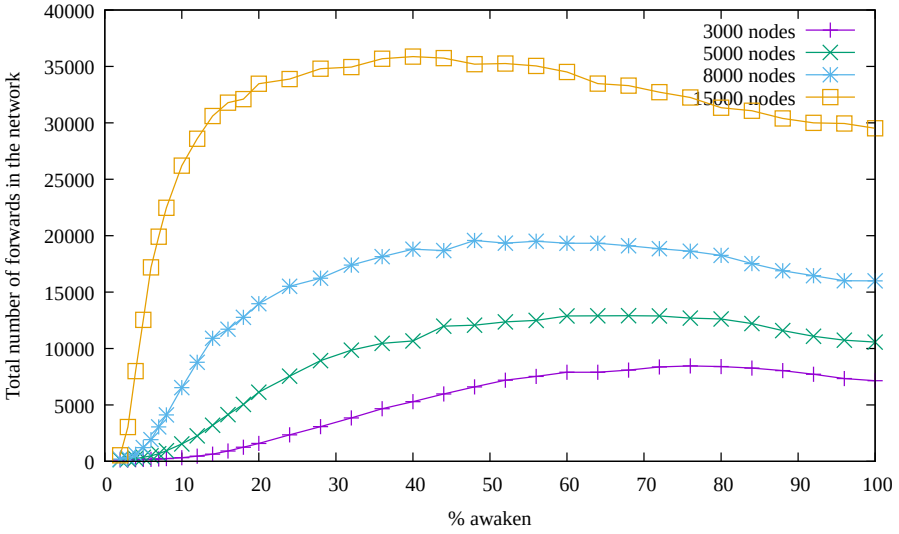


Figure 4.7: Total number of forwards, for one flow and with interval between packets.

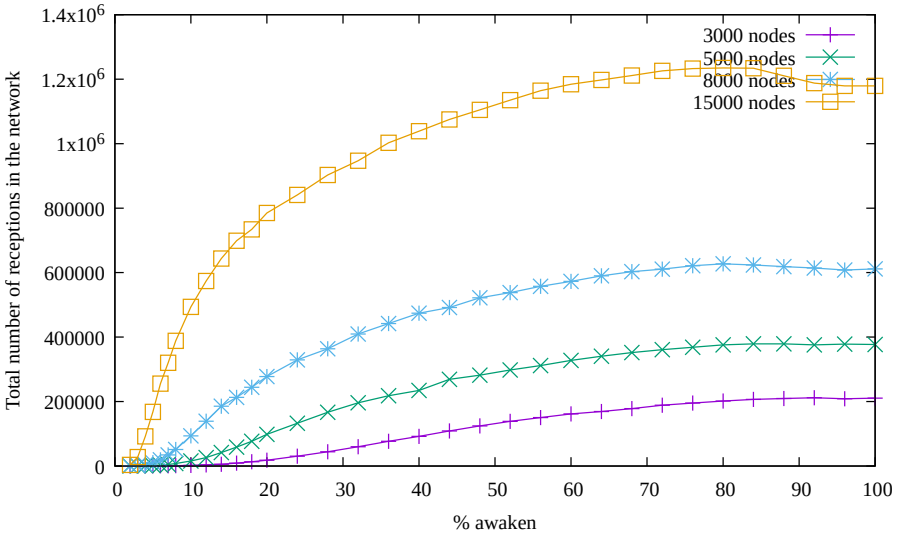


Figure 4.8: Total number of receptions, for one flow and with interval between packets.

4.4.1.4/ RESULTS ANALYSIS: WITH INTERFERING FLOWS

When adding interfering (background) flows in the network, we expect that packets from the main flow have a greater probability of collision, and a greater probability to get ignored by forwarding nodes given that their resources are used for other flows too.

However, Fig. 4.9 shows that the sleep mechanism can improve very significantly the reception rate, e.g. many more packets arrive for 50% than for 100% of awaken nodes.

This ability to improve the usable capacity of the channel is further illustrated in Fig. 4.10. It shows that adding more flows in the network translates into much more forwards (it can be compared to Fig. 4.7), while the reachability stays good.

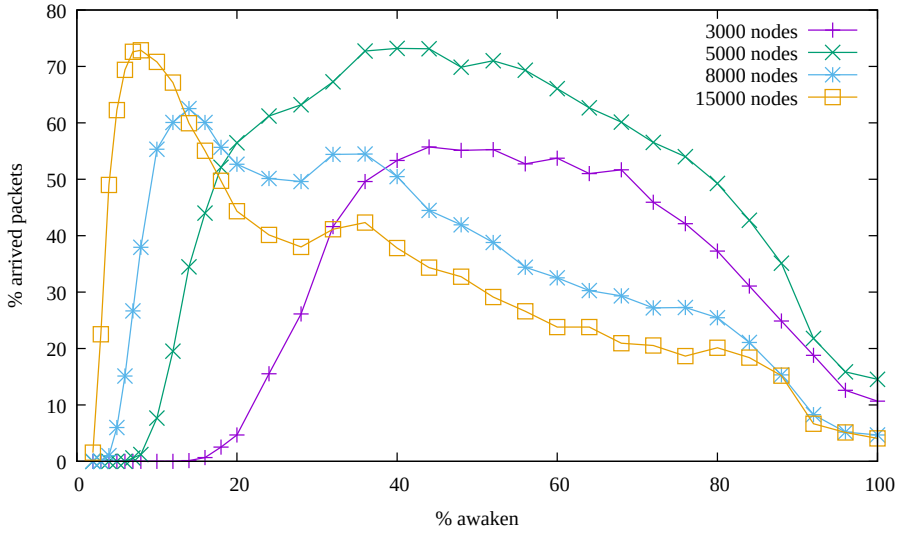


Figure 4.9: Percentage of arrived packets depending on awake time and network density, for 20 concurrent flows and with interval between packets.

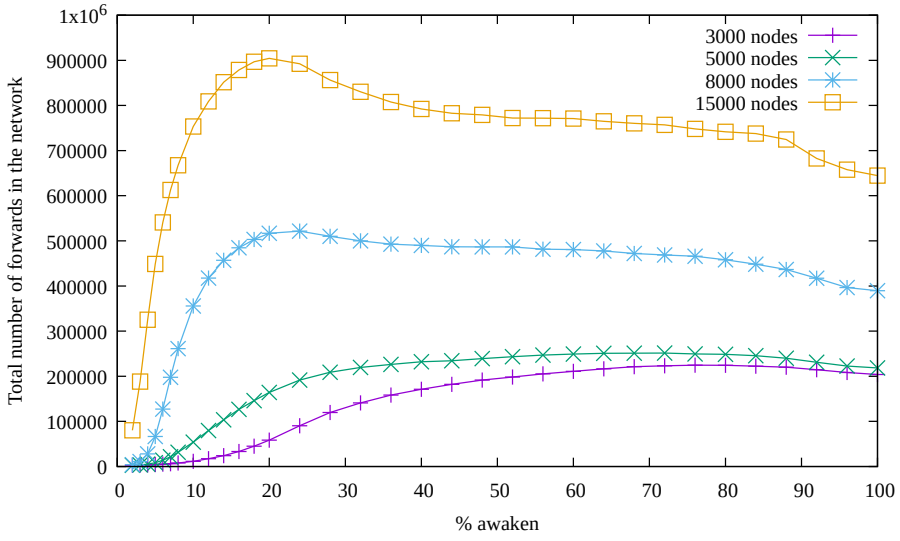


Figure 4.10: Total number of forwards in the network, for 20 concurrent flows and with interval between packets.

4.4.1.5/ CONCLUSION

We discussed the integration of the proposed sleeping mechanism into the SLR routing protocol and in the context of very dense nanonetworks. Evaluations were conducted by using a detailed simulation tool.

Nanonodes are not able to individually process the full throughput of this channel. Especially in dense to very dense environments, when exposed to broadcasting schemes, all nodes in an area tend to reach saturation and start missing (ignoring) new incoming packets.

By allowing an asynchronous sleep mechanism, we are able to improve significantly the amount of data that can be successfully transferred in the network in a given amount of time (we increase the usable capacity). Moreover, as not all nodes participate in all local transmissions anymore (remember that they individually receive much less packets), the load is dispatched. As individual nodes see less activity, they also use less resources (energy, CPU, memory, ...)

Evaluations above clearly show that the percentage of time awake needs tuning by considering the local network density and even the number of locally competing data flows. We have shown that an optimal value can be found, that optimizes the available capacity. The number of packets successfully transmitted increases while the number of ignored packets and collisions does not change much or even decreases. Each application/deployment can benefit very significantly from this tuning.

4.4.2/ Duty-cycling in heterogeneous networks with destination node

In this section we evaluate the usefulness of implementing the sleeping mechanism in heterogeneous networks (different nodes' density in the network), in addition to configure a dynamic awaken duration for nodes based on a density estimation algorithm.

Packet loss at the destination zone might occur, when the packet arrives at its destination, and the intended node is asleep. This section focuses on a special work that should take place only at the destination zone, to increase the chance or packet reception. The proposed algorithm (full retransmission) is already explained in 4.3.2.1.

Nanonetworks have different applications in various fields, extending from environment monitoring, industrial manufacturing, and agriculture to an enormous number of applications in medicine (like drug delivery, diagnostics and surgical operations).

Nowadays, the health sector is one of the most important sectors in which nanotechnology is involved. A nanonetwork can collect vital patient information and provide it to computing systems, providing more accurate and efficient health monitoring. These real applications will mostly be implemented in a heterogeneous nanonetwork due to its structural nature. For that reasons, it is necessary to understand and validate the behaviour of the routing protocol used in places where nodes have different densities.

4.4.2.1/ Network scenario

In our simulations, the network topology consists of a heterogeneous network as a 2D area of size 6 mm x 6 mm, shown in Fig. 4.11. It is composed of three rectangles of size 6 mm x 2 mm, with node average densities of: (rectangle 1: 160, rectangle 2: 80,

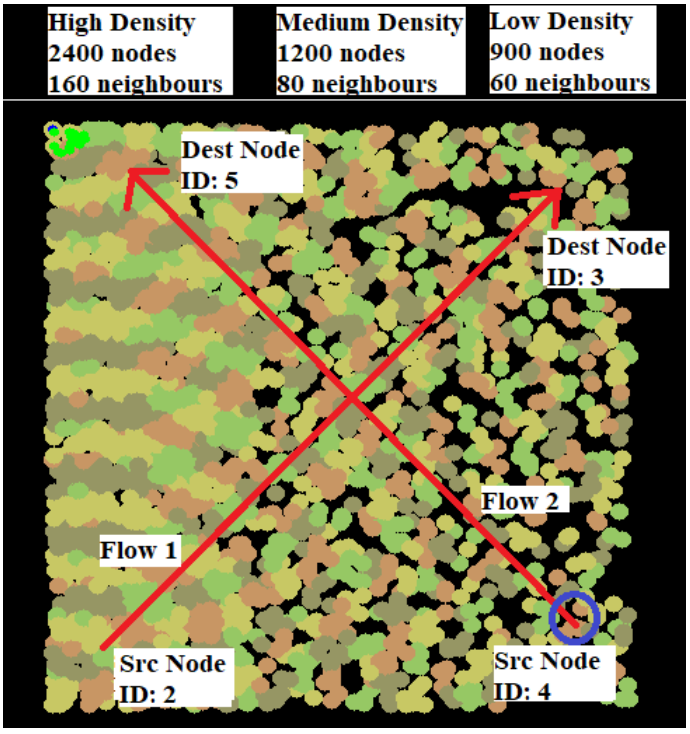


Figure 4.11: The network topology used, with 2 flows and 3 rectangles of different densities.

Table 4.3: Simulation parameters.

Size of simulated network	6 mm * 6 mm
Size of rectangles	6 mm * 2 mm
Number of nodes	4500
Communication radius	500 µm
Hops to reach the furthest node	17
β (time spreading ratio)	1000
T_p	100 fs

and rectangle 3: 60). Nodes are always placed randomly except SLR anchors, sources, and destinations nodes. Source nodes are placed at the bottom of the network near the edges.

Given the environment size and the communication range (0.5 mm), the furthest possible receiver is in the corner at $\sqrt{2} * rectangle\ side/2 = 4.2$ mm away from the source. It means that a packet from the source needs at least 17 hops $((4.2/0.5) * 2 = 17)$ to reach the furthest possible node. Table 4.3 lists the parameters used in the simulations.

There are 2 flows. The source nodes are at the bottom right and bottom left of the network, while the destination nodes are at the top left and top right. Flow 1 traverses the network from left to right through different nodes' density areas (high, then medium, then low), while it is the opposite for flow 2. According to the previous network settings, packet routing will be somewhat complicated and challenging.

The sleeping mechanism is applied to all nodes. To avoid random effects, each point in the following figures represents the average of 10 simulations, each with a different random generator seed to set the beginning of the sleeping period for nodes. In each simulation round, the awaken duration percentage and the number of neighbours are changed, while the other parameters are kept identical.

4.4.2.2/ COMPARISON METRICS

We analyse the sleeping mechanism efficiency using three metrics:

- Packet reachability: the number of packets that have reached the destination node (while applying the full retransmission algorithm). If several copies of the same packet reaches the destination, only one packet is counted.
- Total number of sent packets: the total number of sent packets in the network, for instance in a multi-hop transmission a same packet is sent by several nodes (routers), hence it is counted several times.
- Awaken duration: the interval of time where the node is awake. After this duration, the node goes back to sleep for the rest of the duty cycle (T_s). In BitSimulator, through the command line we can specify the percentage of awaken duration as a variable, or the average density of neighbour nodes (awakenNodes) via DEDeN that calculates it automatically.

4.4.2.3/ RESULTS ANALYSIS: SAME AWAKEN DURATION FOR ALL NODES

We recall that the time between two consecutive bits is $T_s = T_p * \beta$. To determine an awaken percentage for every node in the network (e.g. 20% is equivalent to 20 000 fs) means that all nodes will awake for this percentage in a time duration equal to T_s. Inside the cycle, all the nodes have the same awaken duration (or a percentage of T_s), but the beginning of the awake interval is different for each node and is randomly determined.

Fig. 4.12 shows the average of packet reachability, i.e. at least one packet succeeding in reaching the destination node. We notice that packets are unable to reach the destination for an awaken duration ranging from 5 to 20%. This is due to the insufficient number of awaken nodes especially in low density areas, the flow propagation stops at the beginning. Recall that we are testing in a heterogeneous network, where areas' densities are different.

Increasing the awaken duration immediately increases the average packet reachability. An awaken duration of 70 000 fs allows all packets from flow 1 to be received. Recalling

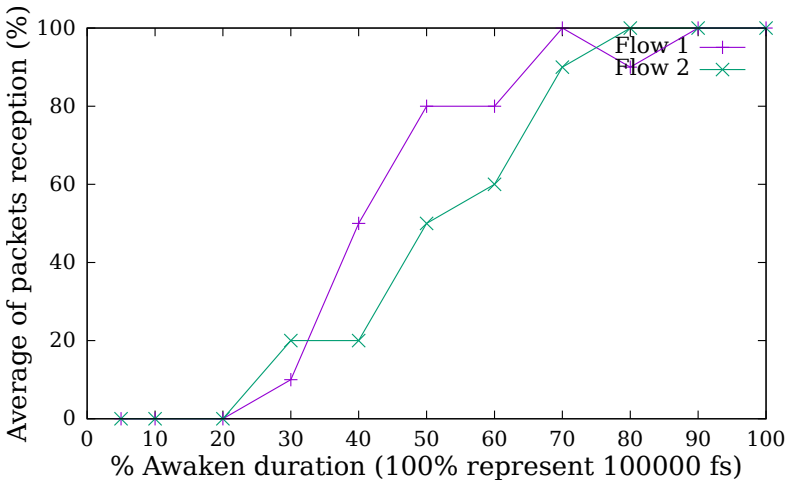


Figure 4.12: Percentage of arrived packets depending on awaken duration, for 2 flows.

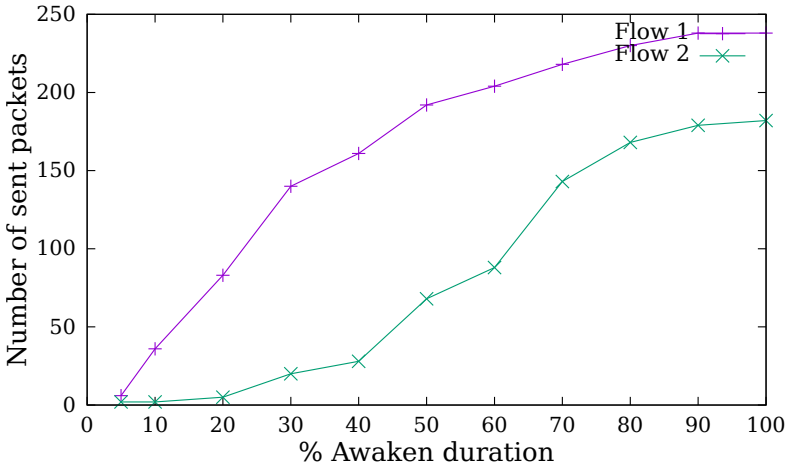


Figure 4.13: The number of packets sent varies with node awaken duration.

that due to the sleeping phenomenon, it is possible for the packet to arrive while the destination node is asleep, and this results in the packet being lost. This is what the curve shows at 80 000 fs (flow 1). For flow 2, an awaken duration of 80 000 fs allows a total packet reception. *Note that 100% of awaken duration means that all nodes in the network are awake all the time.*

Given that the destination is far away from the source, the packet transmission between nodes occurs by hops, so a packet is sent several times, by several nodes in the path. The number of packets sent varies with the node awaken duration. Fig. 4.13 shows that the number of packets sent is very close to 0 at the beginning for both flows, due to the low number of awake nodes. Increasing the awaken duration percentage raises the number of awaken nodes, which results in an increasing number of packets sent too. This number reaches a value of 238 (flow 1) and 179 (flow 2) where the awaken duration is 90%.

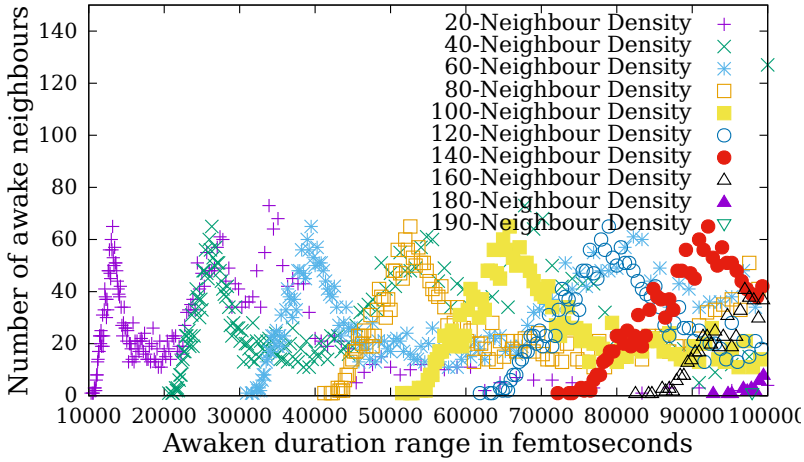


Figure 4.14: Number of awaken nodes in a duration range for different average densities.

4.4.2.4/ RESULTS ANALYSIS: DIFFERENT AWAKEN DURATIONS FOR NODES, BASED ON LOCAL DENSITY

The sleeping mechanism dispatches the load of sent packets among neighbouring nodes. Therefore, taking into account the neighbouring nodes' density to determine the nodes awaken period can be beneficial to the transmission process. Fig. 4.14 shows a disparity in the nodes' awaken duration range. For an average density of 60 neighbours, the node awaken duration ranges between 30 000 fs and 100 000 fs. For a density of 100 neighbours, the node awaken duration range starts at 50 000 fs. As the average of neighbours' nodes increases, the awaken duration range shrinks and goes close to 100 000 fs.

Packet reachability can be significantly improved depending on the average density. For example, when specifying awakenNodes=20 to BitSimulator, each node executes DEDeN, presented in section 2.3 and computes the awake interval according to a simple formula. Starting from 20 as average density, packet reception starts increasing (Fig. 4.15). An average density of 50 neighbours is sufficient for packets to reach their destinations for flow 1, whilst 60 neighbours for flow 2 are needed. Sleeping improves network behaviour by limiting the amount of traffic an individual node can see, but provisions have to be made to ensure the destination node is not sleeping when data packets reach it (example flow 2 at 130 neighbours nodes, where the curve has lost its stability). *Note that, in this scenario, an average density of 190 means that all nodes are awake all the time*.

When average density is low, most packets are not transmitted (because packet propagation stops at the beginning or in their path to the destination). Fig. 4.16 shows that for low average densities (e.g. 20, 30), the number of packets sent is quite low (close to 0). Indeed, with the increasing of the neighbours' nodes average density, packets sent will exponentially increases. For 140 neighbours, the number reaches its highest value at 238 for flow 1 and 182 for flow 2. We notice that with 60 neighbours' nodes (where all

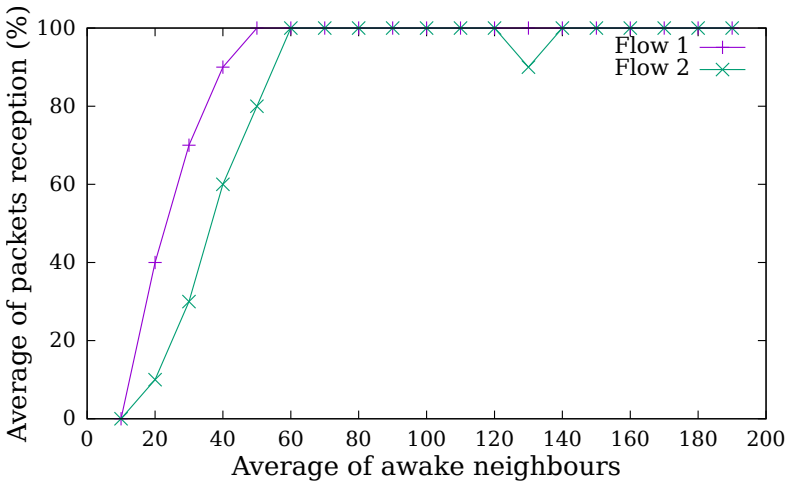


Figure 4.15: Percentage of arrived packets depending on neighbours nodes average, for 2 flows.

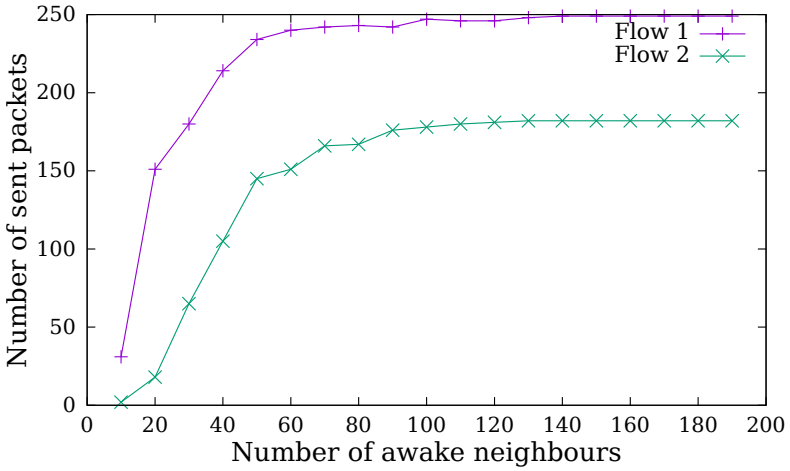


Figure 4.16: Packet transmission cost depending on the average of neighbours' nodes.

packets reach their destination), the total number of packets sent for flow 1 is 205 and 155 for flow 2.

4.4.2.5/ AVERAGE DENSITY VS AWAKEN PERCENTAGE

There are two ways to determine the node awaken duration. Either static (known as awakenDuration) or dynamic (known as awakenNodes) based on the node's average density. A static duration means that all nodes have the same duration inside T_s. Contrarily, with dynamic based, nodes have different awaken duration calculated based on the neighbours' nodes density.

A neighbour nodes density estimator (DEDeN) allows a node to estimate the number of its neighbours. Based on this value, nodes will set their awaken interval (awaken duration)

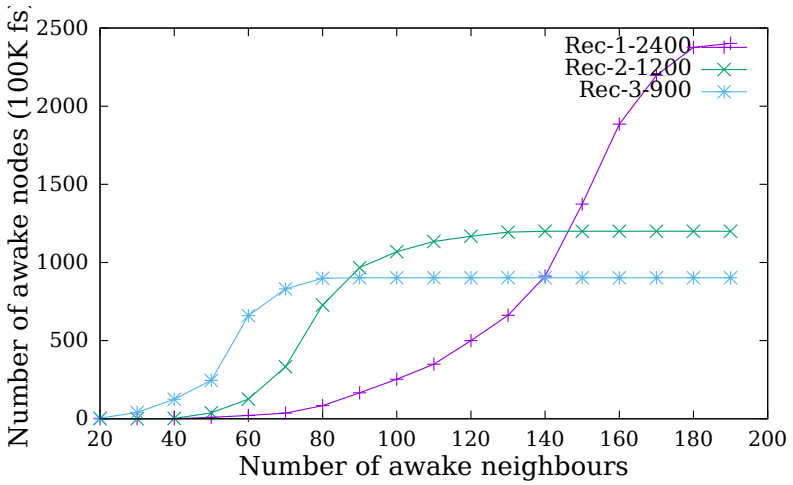


Figure 4.17: Number of full-awake nodes over different average density values.

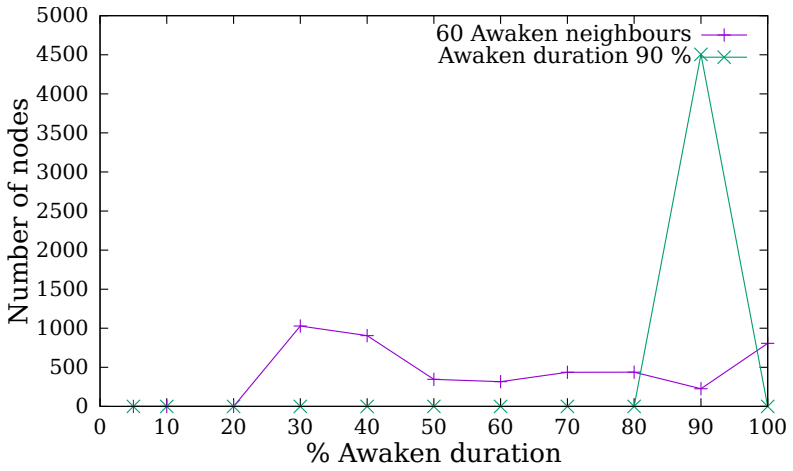


Figure 4.18: Number of nodes depending on an awaken duration of 90% and 60 neighbour nodes.

(Fig. 4.17). Therefore, the distribution in nodes awaken duration will be clearly shown in the heterogeneous network because areas in this type of network are of different densities compared to a homogeneous network. While fixing an awaken percentage (e.g. 80%), means that all the nodes in the network will be awake for the same duration (e.g. 80 000 fs).

The longer the node sleeps, the lower the consumed resources. An average density of 60 neighbours shows an awaken duration distribution (less resource consumption) ranging between 20 000 and 100 000 fs (different awaken duration for each node). Furthermore, for an awaken percentage of 90%, all the nodes in the network (4500 nodes) will be awake along this duration (Fig. 4.18).

The total number of sent packets is a metric we rely on to compare between the methods used in applying the sleep mechanism (awakenDuration vs awakenNodes). Based on the

previous results, setting the node awaken duration based on node neighbour density (using DEDeN) will help in decreasing the number of packets sent by 14% when compared to a fixed awaken percentage (awakenDuration) based on Fig. 4.16.

To conclude, in heterogeneous networks, the SLR routing protocol is acting well when crossing areas of several densities. It smoothly deals with nodes of different awaken durations, and packets are successfully delivered to their intended destination. Applying DEDeN allows nodes to have different awaken durations compared to a static awakenDuration. This leads to a decrease in packet transmission cost, and preservation of node resources (CPU, energy, memory), therefore an increase in network lifetime.

4.4.2.6/ RESULTS ANALYSIS: PROCESSING AT THE DESTINATION ZONE

As explained before, the proposed sleeping mechanism reduces the congestion problem and preserves node resource consumption along the path from the source to the destination. But the very definition of destination may change depending on the application. If the destination is defined as an SLR address, it means that we want the packet to reach this SLR zone and that at least one node in this zone must receive this packet. In that case, the mechanism we proposed is efficient and can be used as is. On the other hand, if we aim to reach a specific node at the destination zone, then more aspects have to be taken into consideration. When a packet arrives at the destination zone, the destination node by chance may be asleep and consequently misses the packet. Different strategies might be used to solve this problem, depending on the node peculiarities, the local density and the application requirement. The algorithm is well explained in Section 4.3.

Fig. 4.19 shows the impact of applying our proposed full retransmission algorithm at the destination zone. The algorithm clearly enhances the average of packet reception with the flow 1-patch compared to flow1-nopatch (without the algorithm). A value of 40 neighbours nodes ensures an average of 100% of packet reception by the destination node. Indeed, the positive impact is also clearly shown for flow 2-patch at the 130 of awake neighbours, comparing to the flow 2-nopatch, where the destination node missed the packet because it was asleep. The proposed algorithm shows its effect in increasing node chances in receiving the packet if it was asleep at the moment the packet reaches its zone (Fig. 4.20).

4.4.2.7/ CONCLUSION

We have covered the effect of the sleeping mechanism in a heterogeneous nanonetwork, in addition to an algorithm that aims to solve the data loss problem in case the node is asleep when the packet arrives at the destination zone.

The new idea embeds a density estimator algorithm (DEDeN) to automatically tune the

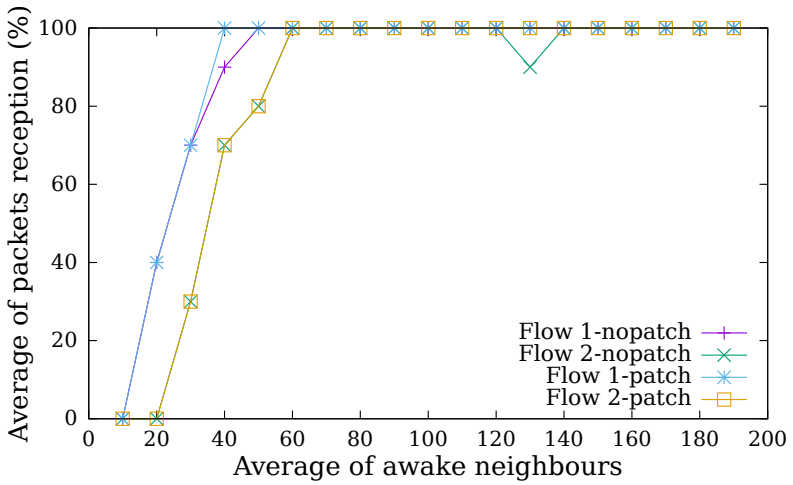


Figure 4.19: The impact of the retransmission algorithm.

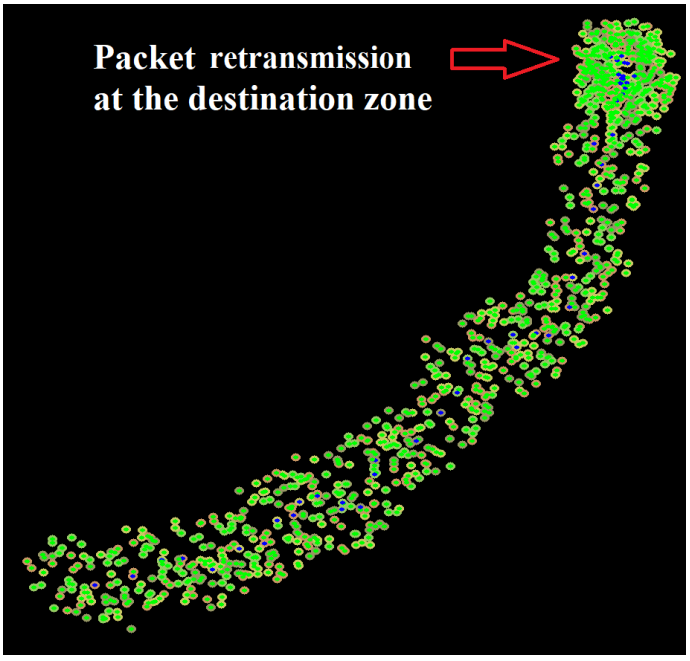


Figure 4.20: The retransmission algorithm applied at the destination zone increases the number of exchanged packets in that zone.

node awaken interval (awakenDuration). The estimator algorithm can be executed in different times (network deployment, or when a node needs to estimate its number of neighbours). It shows its usefulness when used in conjunction with the sleeping mechanism. It becomes clear that relying on average neighbour node density preserves node resources (CPU, energy, memory ...) and decreases the number of sent packets comparing to results based on static percentage of awaken duration.

Considerations have also been taken for the specific case of the destination zone (packet loss by the destination node because its sleep may coincide with the packet's arrival at the destination zone). An algorithm has been proposed in case the application needs data

packets to reach a specific node in the destination zone. Where the node that receives the packet at the destination zone will retransmit the packet again at the end of its awaken duration.

4.4.3/ PROBABILISTIC PACKET RETRANSMISSION WITH DESTINATION NODE

In this section, we evaluate the enhanced retransmission algorithm presented in section 4.3.2.2 used by the nodes in the destination zone. This algorithm increases the chance of a destination node capturing the intended packet while decreasing the number of participating nodes in the retransmission process.

4.4.3.1/ NETWORK SCENARIO

In our simulations, the network topology consists of a homogeneous network as a 2D area of size 6 mm * 6 mm, cf. Table 4.4. One packet traverses the network; the source node is at the bottom left of the network, while the destination node is at the top right, cf. Fig. 4.21.

Parameter	Value
Size of simulated network	6 mm * 6 mm
Number of nodes	25 000
Communication radius	500 µm
Hops to reach the furthest node	17
AwakenDuration	6000 fs
T_p	100 fs
β (spreading ratio)	1000
Packet size	1000 bit

Table 4.4: Simulation parameters.

In the following analysis, all the nodes use the sleeping mechanism. The simulation is repeated several times by changing the node awaken duration percentage, all the other parameters being kept identical.

The metrics used to analyze the algorithm efficiency are the number of nodes that retransmit the packet at the destination zone, and the reliability of receiving at least 1 copy of the packet by the destination node.

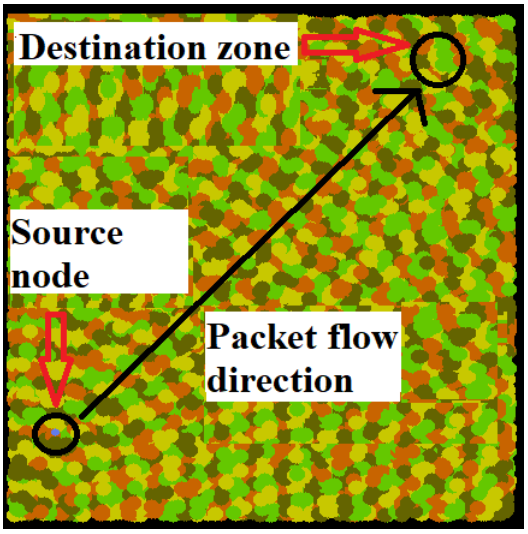


Figure 4.21: The network used to evaluate the enhanced retransmission algorithm.

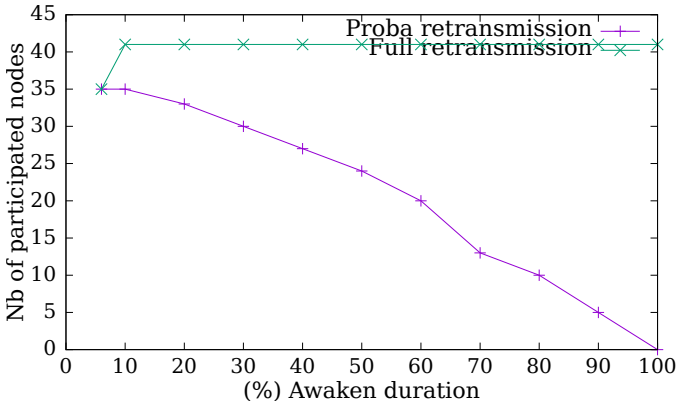


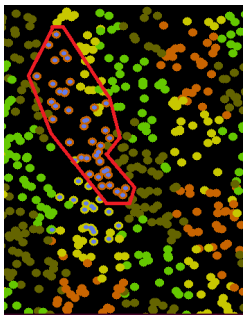
Figure 4.22: Participating nodes in packet retransmission at the destination zone.

4.4.3.2/ RESULT ANALYSIS

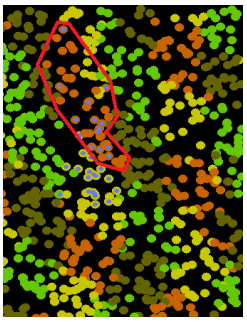
We recall that a static awaken percentage for every node in the network (e.g. 20% is equivalent to 20 000 fs) means that all nodes will be awake for this percentage in a time duration equal to T_s. Also, inside this cycle, all the nodes have the same awaken duration (or a percentage of T_s), but the beginning of the awake interval is different for each node and is randomly determined. It is useful to note that the purpose of using the probabilistic packet retransmission is to decrease the number of participating nodes in packet retransmission at the destination zone while ensuring the packet receiving by the intended node.

We start comparing the probabilistic algorithm with the full retransmission. We notice that for an awaken duration of 6%, both transmission mechanisms have the same number of participating nodes (35) 4.22. This is expected since for a low awaken duration the probability of retransmission is high.

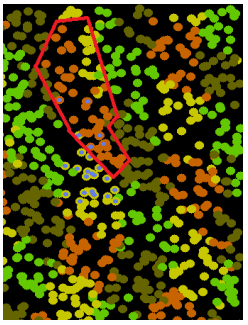
The relation between the awaken duration and the probability is inversely proportional.



(a) Full retransmission, all nodes paticipating at the destination zone.



(b) Proba retransmission applied for 70% of node awaken duration.



(c) Proba retransmission applied for 90% of node awaken duration.

Figure 4.23: VisualTracer sketch for the destination zone, for the number of participated nodes with full and proba packet retransmission.

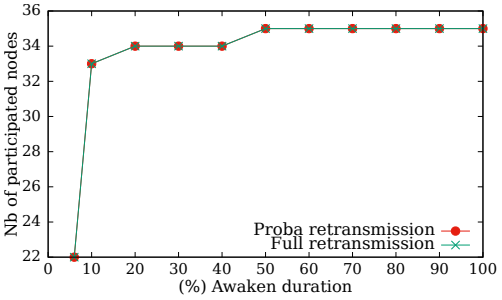
While the awaken duration increases, the probability decreases, therefore it reflects a decrease in the number of participating nodes while using the probabilistic retransmission algorithm. For example, for an awaken duration of 50%, the number of participating nodes is 25 with the proba algorithm, while with the full retransmission this number remains steady, 41 nodes, for all the simulated awaken duration.

Fig. 4.23 is a sketch extracted from VisualTracer. The figure shows the benefit of applying the probabilistic retransmission instead of the full retransmission. (a) Shows that all nodes (41) participate in retransmission at the destination zone. Applying the proba retransmission in (b) while using the same awaken nodes percentage, shows a decrease of 68% of participating nodes. An increase in the awaken duration percentage (c) shows a higher decrease in the number of participating nodes (88%) compared to the full retransmission.

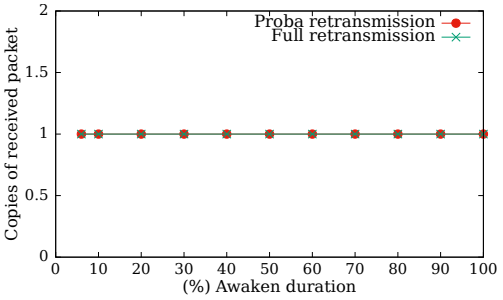
It is important to ensure that the algorithm does not affect packet routing in the previous zones. Fig. 4.24a shows that the number of participating nodes in the packet routing does not change while applying the retransmission algorithm. Therefore, the previous zone is just playing the role of routing the packet and sending it to the next hop (*zone*).

Fig. 4.24b shows the reliability of packet receiving by the destination node. In all simulations, and using several awaken duration percentages, the destination node is still able to receive at least 1 copy of the intended packet.

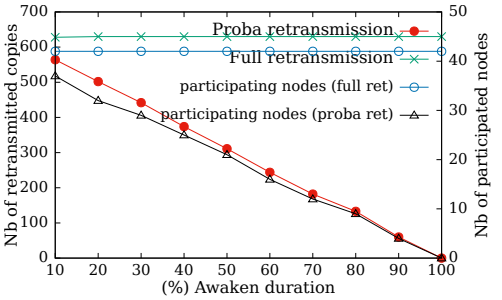
Several flows in a network might affect the algorithm being applied. For that reason, it is necessary to evaluate our algorithm when using several flows too. Fig. 4.24c depicts the results of a simulation with 15 flows from 15 source nodes sending packets to one destination node. For full retransmission, the number of retransmitted packet copies ($\approx$620) stays stable along all the awaken duration percentages used. Even if this number decreases, it achieves around 50% fewer retransmissions ($\approx$310) at 50% of node awaken duration, and 11% fewer retransmissions at 90% of awaken duration.



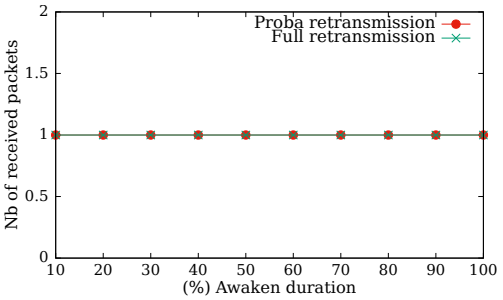
(a) Retransmission mechanism does not affect the previous zone.



(b) The reliability of receiving at least 1 copy of the packet by the destination node.



(c) Number of retransmitted packet copies and participating nodes.



(d) The destination node receives at least 1 packet copy of each flow.

Figure 4.24: Probabilistic retransmission simulations results.

This is also reflected in the number of participating nodes, where all the nodes (42 nodes) are participating in the full retransmission. This number decreases according to the awaken duration percentage used in the proba retransmission (e.g. 20 nodes at 50% of awaken duration, and 5 nodes at 90% of awaken duration).

Fig. 4.24d shows that the use of the probabilistic retransmission algorithm contributes to enhancing packet reception reliability: the destination node receives at least 1 copy of the retransmitted packet from each flow. Once again, the probabilistic algorithm proves its effectiveness even in case of several flows.

4.4.3.3/ CONCLUSION

We proposed and discussed a probabilistic retransmission algorithm at the destination zone where not all nodes participate in the retransmission process. The goal is to ensure that the destination node receives the packet, while reducing the number of packets exchanged. In the proposed algorithm, the number of participating nodes is correlated to the percentage of nodes awaken duration.

The simulation results show that the destination node is still able to receive the intended packet while decreasing the number of retransmissions at the destination zone.

4.5/ SUMMARY

In any network, the routing and congestion control protocols play a key role in determining network behavior. The routing protocol designs the route from the source to the destination, to allow packets to reach their intended destination. Nanonodes are equipped with limited resources (small CPU, small memory, and low battery). Therefore, it is important to find a way to preserve nodes' resources, therefore, extend the network lifetime. Due to the differences in terms of hardware capabilities, the traditional algorithms in macroscale network is not adequate for nanonetwork.

Based on the above reasons, we proposed an innovative sleeping mechanism that contributes to preserving nodes' resources and decreasing the congestion in the network during the packets routing phase. It's important to note that applying sleeping mechanism does not mean to shutdown the node and then wake-up again. Instead, during the sleeping duration, the node will ignore the received packets. This will lead to not consuming resources such as processing, energy, or memory.

Based on the results described in the evaluation section, the SLR routing protocol embedded with sleeping mechanism shows its effectiveness in homogeneous and heterogeneous networks. The duty-cycling technique succeeded in decreasing the number of forwarders along the path from the source to the destination.

Another factor that should be taken into consideration, is the reliability of packet reception. With the use of the sleep algorithm, a problem arises in the possibility that a node within the destination area could not receive the packet because its sleep duration coincided with the packet's arrival. Therefore, the node misses the packet. The majority of the time, once the packet reaches the destination area, the task is considered successful because any node that got the packet can provide the service. The problem appears when a specific node at the destination area must receive the packet to provide the service.

To maintain reliable packet receptions, a full and probabilistic retransmission algorithm was proposed. It is important to note that the algorithm will only be executed in the destination region. Therefore, it will not be a load for nodes' and it will not be a cause of resource consumption. Both algorithms show their effectiveness in allowing the intended node to receive the packet. But the probabilistic algorithm is the improved version, where not all nodes participate in the retransmission process. This will back with a benefit in preserving nodes' resources at the destination area, as well as reducing the network congestion there.

The final result is a routing protocol, occupied with a duty-cycling technique, with a static or automated node's awaken duration, while maintaining high reliability of packet receptions by the destination node.

5

IMPROVED NANO-SLEEPING USING A COUNTER-BASED ROUTING

Due to the chronological process of my PhD thesis, it was necessary to start proposing a main idea to solve the problem of resource consumption in nanonetwork communication. Starting with the sleeping mechanism, it shows its effectiveness in reducing the number of forwarder nodes, along the routing path from the source node toward the destination zone/node.

Deeply analyzing the evaluation results of sleeping mechanism shows that only decreasing the number of forwarders is not enough to reach the optimal solution. Additional work should be done to fine tuning the sleeping mechanism and reach our goal. Therefore, we have moved to adopt a gradual approach to reveal, analyze and fix one by one the routing inefficiencies to improve nano-sleeping.

Finally, we will conclude with a robust scheme with all needed improvements that guarantee drastically decreased forwarders and reception nodes too, in addition to applying an efficient algorithm of packet retransmission at the destination zone to guarantee high reliability in packet reception.

5.1/ PROBLEM STATEMENT

Improving the behavior and channel usage of multi-hop nanonetworks is a complex problem. We highlight the problem we want to solve through multiple variations of a specific multi-hop scenario. At first, we only use the SLR routing protocol and observe that numerous nodes either directly on the route or close to it receive the routed packet. This is considered bad, as we remind that the goal of our work is to reduce the average load on nodes by reducing the number of receivers of any given forwarded packet while maintaining an end-to-end packet delivery ratio.

We then complement SLR with the fine-grained sleeping mechanism. It shows its effec-

tiveness in reducing the number of forwarders section 4.5, but it fails to reduce the number of receivers, which is not our aim.

In the coming section, we will discuss two serious drawbacks of SLR with sleep mechanism: the spreading phenomena, and the packet propagation stop at the source.

5.1.1/ SLR PACKET RECEPTIONS PROBLEMS WITH SLEEP

As explained before, the end goal of the sleep mechanism is to reduce the cost of forwarding data packets by *reducing the number of receptions*, which in turn should also reduce the number of retransmissions (like many other protocols). However, simply applying sleep to SLR is not sufficient. In the following sections, a detailed explanation of the two negative phenomena (spreading and propagation stop at the source) reasons are described.

5.1.1.1/ SPREADING

In this section, we describe in detail the reason for the packet spreading phenomena, it's a great influence on the number of receptions and its impact on the network behavior.

We use the following scenario to explain this phenomenon:

Number of nodes	25 000, randomly distributed
Awake duration	6000 fs (6% of T_s)
Backoff window	20 000 fs (= 200 T_p)
Data flow	One packet sent from one side of the area to the other

Figure 5.1 graphically shows that few nodes (in green) close to the source receive the packet, but their number increases afterward, up to the point where, surprisingly, all nodes on the path participate. This is contrary to our goal of reducing the number of nodes participating in the routing.

The explanation of this phenomenon is the following. In a dense network, when a node sends a packet, numerous nodes receive it at almost the same time (propagation time is comparable to pulse length T_p) and forward it. The usual method to avoid concurrent packet transmission and hence packet collisions is to make nodes wait for a small random duration, known as backoff.

The backoff greatly affects the sleep mechanism. Even if this backoff is small, the copies will be received by the next nodes in a time window that will only increase hop after hop. As the multiple copies transmissions are scattered over larger and larger windows, they will reach more and more nodes that may be awake only for a part of those windows.

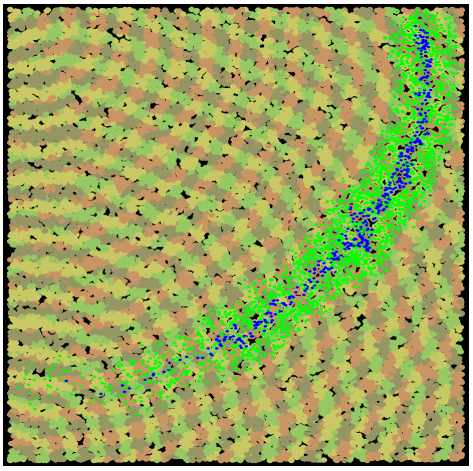


Figure 5.1: Effect of the spreading: the number of receivers, in green (and thus forwarders too, in blue) increases with the number of hops from the source (at bottom-left).

Should the backoff be large enough, or the path long enough, packet forwarding will eventually be spread over a window equal or larger than T_s. At this point, if the number of transmitting nodes is high enough, at each hop all neighbouring nodes receive a copy of the packet. This happen even if they are awake only for a fraction of each T_s, which is exactly what the sleep mechanism wants to avoid.

The spreading can be formalised as follows. Relative to the transmitting time, the forwarding occurs at $[p, p + w]$ later, where p is the packet propagation delay and w the backoff window (from which the backoff is chosen randomly). p depends on the distance between sender and receiver, however due to the short distances (e.g. $p = 2T_p$) and to the destination-oriented routing, p is negligible, so the spreading at first hop at worst is w. Generally, at the Nth hop, the spread is at worst Nw. For instance, with values $w = 200\,T_p$ and $\beta = 1000$ (cf. Table 5.2), after almost $\beta T_p/w = 5$ hops the spread might span over the whole T_s. After this point all the nodes receive a copy of the packet, hence the sleep mechanism becomes useless. This confirms the result of Figure 5.1.

To conclude, the spreading makes it so that more and more (and eventually all) the nodes down the path receive the packet. This is exactly what the sleep mechanism tries to avoid. Thus, as a universal rule, the spreading must be avoided.

5.1.1.2/ PROPAGATION STOP AT THE SOURCE

The normal result of a communication is to deliver the packets to their destination. In this section, we describe in detail the reason of packet interruption, which is represented by the stop of packet propagation at the source. However, when simulating long multi-hop paths, we noticed that many packets did not reach the destination zone. We proposed a suitable scenario to find out the cause of the packet stops. The scenario parameters are

Number of nodes	15 000, randomly distributed
Awake duration	7000 fs (7% of T_s)
Data flow	100 unique packets crossing most of the network

Table 5.1: Simulation parameters used to highlight packet loses near the source

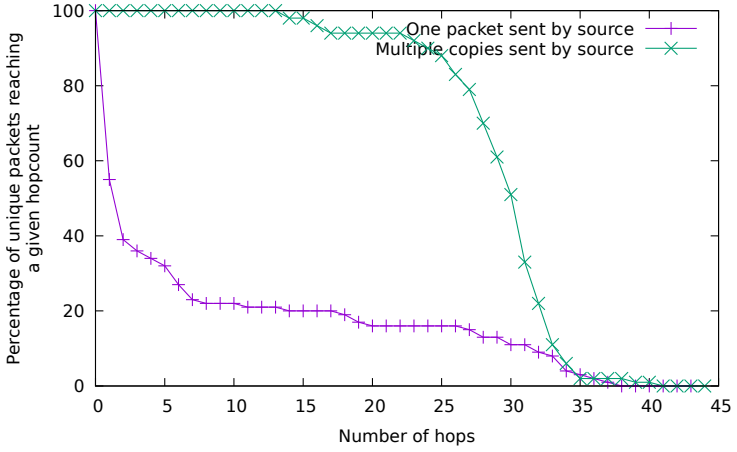


Figure 5.2: Number of losses as a function of hops, for one and multiple packets.

listed in Table 5.1. In this scenario, over some time period, the source send 100 unique data packets toward a distant destination.

Only counting packets reaching the destination is not enough here. We need to know up to which distance packets went before being lost. Figure 5.2 (violet curve) plots the number of unique packets that reached a given hop count. It shows that almost half of the packets are lost at the very first hop, and the rate of losses progressively slows down after that. So when the initial source sends a packet, often none of its neighbours receives it, because all of them are sleeping at the time the packet is transmitted.

The reason this happens at source and not later on the path resides in the different number of forwarders at source and during routing. At the sender zone only one packet is transmitted (by the source); if awake duration is small, in average only a few nodes are awake at a given time, and there is a relatively high probability of no node being awake when this packet is sent, thus stopping the propagation. Contrarily to this, during routing, usually more than one node forwards the packet, hence several copies are transmitted, hence the probability of propagation stop is smaller.

A simple method to avoid the propagation stop at the source is to make the source send multiple copies of the same packet, so that the probability of reception by the neighbours of the source increases. Ideally, the number of copies should be similar to the number of copies sent along the rest of the path, so that the same probability applies both to the source and during routing. This is illustrated on the green curve of Figure 5.2: compared

to the one packet at source curve, the sharp decrease at the beginning has disappeared.

Indeed, sending multiple copies of the same packet at the source is considered a solution. But this method must be avoided because the overall number of receptions and transmissions will increases significantly. Here, the subtlety resides in the exact time to send the copies (the absolute time when the copies are sent does not matter, but only the value modulo T_s). Making the source send the copies with an inter-copy time either small, or spread over the whole T_s does solve the problem near the source, but has catastrophic results during routing.

Solving the problem at the source by sending multiple copies is essentially a forced spread. In terms of retransmissions, it behaves exactly as the problem stated in section 5.1.1.1. So, it is important to have a solution that does not affect the number of retransmissions in the network and does not exhaust nodes' resources.

5.1.2/ PROPAGATION MODEL

In real life, due to fluctuations in the environment, nodes may not be able to consistently receive all packets. This is especially true for nodes close to the border of the sender's communication range.

In BitSimulator (the simulator we use), nodes assume an omnidirectional antenna and thus a circular communication range. However, a propagation model more realistic than the simple unit disc (all or nothing) is also used. But we still miss the fact of the reception decreases at the edge of the communication range.

The shadowing propagation model used in this work aims to mimic the usual decrease in the reception rate as the distance to the sender increases, while keeping the very low computational cost required for the scalability. Like in ns2 and ns3 network simulators, this model uses randomness near the edge of the communication range to progressively increase the odd of not receiving a packet as the distance increases.

Figure 5.3 shows which nodes received a given packet in an environment where nodes are homogeneously distributed. In the middle part of the circle, all nodes where able to receive. But as we go further from the transmitter (at the center), we observe a decreasing number of receptions, up to the limit of the communication range, beyond which no more receptions are possible.

We use the same communication range with shadowing as we would have without it. This affects our results in the following ways: (1) For a given transmission, the number of receivers becomes way smaller than its number of physical neighbours (nodes in its communication range). This significantly reduces the overall number of reception events in a scenario. (2) This reduced number of receptions, especially those close to the border

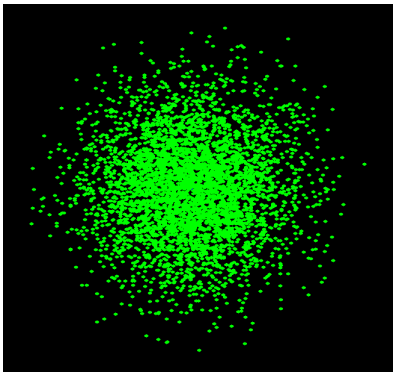


Figure 5.3: Shadowing propagation model, where nodes closer to the edge of the communication range have a smaller probability to receive.

of the communication range affects the initialization phase of the SLR protocol, making its zones smaller and more deformed. The shadowing propagation model hence impacts the routing itself.

5.1.3/ PROBLEM OF NUMEROUS RECEPTIONS

The work relies on three characteristics, which influence receptions: high network density, large bandwidth, and limited nanonode capacity.

High network density means that nodes have a high number of neighbours. In a wireless network, a transmission is by nature a broadcast to the whole communication range. This causes a lot of receptions events, where nodes have to choose what to do with this packet.

In nanonetworks the channel capacity is also very large, in the order of Tb/s. Nanonodes could not process a continuous flow of packets coming at that speed. Not only nodes could not process all these packets, but, in TS-OOK, packets can also overlap without colliding, so that nodes receive numerous packets in parallel. Nanonode have limited memory, energy, processing capacity, among others Canovas-Carrasco et al. (2016).

The theoretical number of concurrent packets is tied to the value of the β parameter. Even with some provision margins when receiving a pulse, a rather conservative value such as $\beta = 1000$ already leads to potentially hundred of concurrent packets on the channel, which is way beyond the processing capacity of a nanonode. Limitations of nanonodes thus can be integrated into one parameter, Maximum Concurrent Receptions (MCR) Arrabal et al. (2019), that makes nodes ignore all packets beyond its value.

Formally, we can consider data broadcasting in a multi-hop network with a node density of ρ (i.e. each node has ρ neighbours). Each new data packet sent will be receive by all neighbours, that will potentially retransmit it. In the next round, nodes will be received

up to $r = \rho$ identical packets to process. The traditional way to reduce the number of receptions r is to reduce the number of retransmitters through various approaches. In a probabilistic approach for example, if only a fraction $k < 1$ of neighbouring nodes forward the packet, then each node receives $r = k\rho$ packets.

We propose a set of mechanisms relying on duty-cycling to further reduce the number of packets to process. Nodes receive and process incoming packets only if they are awake. Thanks to this mechanism, in average, nodes receive only a fraction of the packets. In awake state, they receive an average of $r = dk\rho$, where $d < 1$ is the duty-cycle, i.e. the percentage of time a node is awake, and $k < 1$ is the fraction of nodes retransmitting packets mentioned above.

To conclude, the combination of dense networks (a node has numerous neighbours, and in wireless all of them receive its packet), large bandwidth (at the order of Tb/s), and capacity-limited nanonodes (can process only a few packets at a time) make the number of receptions of paramount importance and call for methods on reducing it.

5.1.4/ PROBLEM OF SLR ON RECEPTIONS

SLR is a destination-oriented geographically restricted flooding. Thus, when the density is high, many nodes can be on the path and consequently send a lot of copies. This affects packet processing and hop to hop routing, detailed in the following.

Figure 5.4 shows an example of SLR routing one packet from a source node (situated near the bottom-left of the area) to a destination (near the top-right) in the following scenario:

Number of nodes	39 000, randomly distributed
Data flow	One packet

All the four sub-figures show the wave patterns that materialize the distance (and thus the coordinates) from the two anchors nodes (situated at the top-right and bottom-right corners of the area).

The top-left sub-figure shows in blue the nodes that consider themselves on the SLR path and retransmit (only once) that data packet when they receive it. Because in this scenario the network is quite dense (30 000 nodes), the retransmitting nodes are also numerous. Here, the packet is retransmitted 1714 times.

The sub-figure in the middle shows in green the nodes that received at least one copy of the packet. Because of the transmission range of the nodes, the area where the packet has been received is wider than the SLR path. Moreover, most nodes in the green area have received the same packet from many different retransmitters. In this scenario, the data packet has been *successfully* received more than 91 000 times!

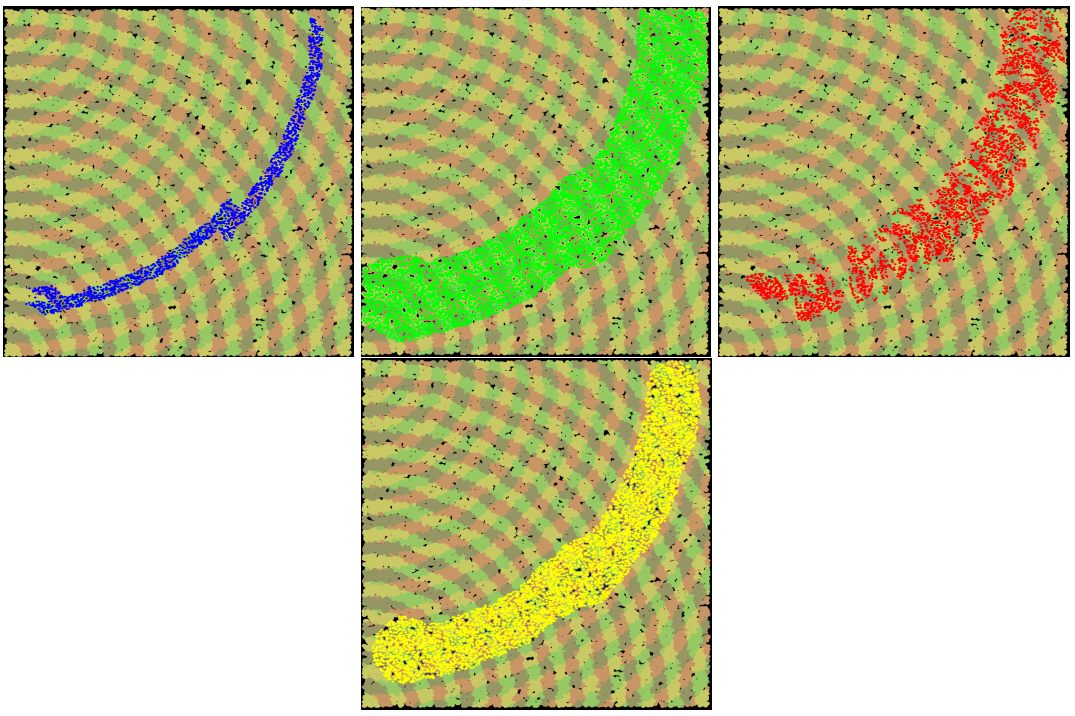


Figure 5.4: SLR routing protocol behaviour: TX nodes (top-left), RX nodes (middle), nodes with collision (bottom) and nodes with ignore (top-right).

Many copies sent also implies a risk of collisions. When packets sent by different re-transmitters arrive at the same time at a node, they may collide. Note that collisions are not systematic between concurrent packets due to the pulse based modulation. In red on top-right sub-figure are represented nodes that experienced at least one destructive collision. Many collisions occur along the path. In this example, the transmission of the packet caused more than 5000 collisions.

The TS-OOK modulation allows for multiple packets to co-exist on the channel at the same time by having their bits interleaved in time. The nature of this modulation allows nodes to correctly receive multiple packets at the same time. However, we recognize that whereas the channel capacity is extremely high, a single tiny node has limited capabilities and cannot process too many concurrent packets (see previous section). New packets arriving while all the resources are already in use can only be silently ignored by a node. This is different from a collision, as the node is not even aware of the propagation of a new packet in the channel. In this scenario, we limit the number of possible concurrent receptions by a node to $MCR = 5$. The sub-figure on the bottom-right show in yellow the nodes that ignored at least one arriving packet. Due to the density of the network and the tendency of multiple neighbouring nodes to receive and then decide to forward a packet almost at the same time, ignored packets are numerous. There are more than 1 000 000 ignored packets in this scenario!

Parameter	Value
Area size	6×6 mm
Communication range	0.5 mm
T_p	100 fs
β	1000
Routing protocol	SLR

Table 5.2: The scenario parameters used.

In the previous sections, we detailed the problems that would increase the number of forwarders and receptions along the routing path. To conclude, the sleep mechanism applied to SLR does not reduce the number of receivers. In section 5.3 we propose successive modifications that pile one over the other to make this combination efficient in terms of the number of receivers. In the following sections, we present the improvements that we have suggested, to achieve an efficient SLR protocol with a reliable sleeping mechanism.

5.2/ NETWORK SCENARIOS

The physical low-level parameters of the scenarios used, are shown in table 5.2. The area side (6 mm) is 12 times greater than the communication range (0.5 mm), which yields several hops between source and destination, allowing to correctly test routing protocol functioning. We remind that β and T_p have the values proposed in the literature.

Beyond those base parameters that stay the same, multiple scenario variants are used in this work. They are required to expose multiple aspects of the considered protocols.

For each of them, we will mention the number of nodes, the number of packets, the awake duration and other parameters of interest.

5.3/ IMPROVEMENT PHASES (EVALUATION AND RESULT ANALYSIS)

5.3.1/ SLR IMPROVED WITH COUNTER-BASED FORWARDING

Recall that the sleep mechanism alone does not achieve the desired result: either the propagation often stops at the source section 5.1.1.2, or the overall number of nodes receiving the packet is too high section 5.1.3, making the sleep mechanism useless. On the one hand, a small awake duration increases the risk to lose a packet because of no node being awake at the reception time. On the other hand, an awake duration large enough to alleviate problem at the source leads, along the path, to a ineluctable increase in retransmissions.

Awake duration	15 000 fs (15% of T_s)
Backoff window	1 000 000 000 fs (= 10 000 T_s)

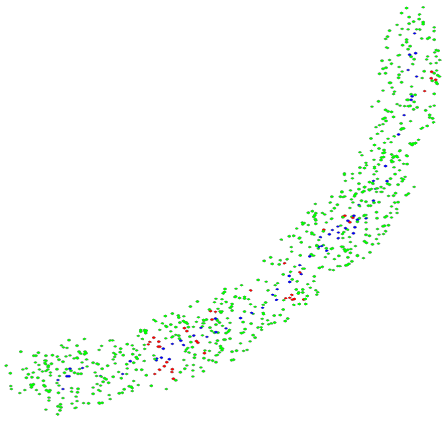


Figure 5.5: A counter-based forwarding with backoff solves both spreading and propagation stop at source.

To solve this apparently unsolvable problem, we propose to supplement SLR with a counter-based mechanism. When a node receives a packet and intends to forward it, it does not do it immediately, but keeps it for some time (backoff); if after this duration, the node has not seen enough copies from downstream nodes (towards destination, as provided by SLR), it forwards the packet, otherwise it drops it.

The backoff needs to be chosen carefully. First, it must be larger than the transmission time of a packet. This is because the channel allows concurrent transmissions, and a packet has to be completely received to be recognized (or, at the very least, its header). If nodes wait for a too short duration before forwarding, they will transmit too early, not being able to take into account copies sent in parallel by their neighbours. The backoff needs also to be random (to avoid collisions between copies). Finally, it must avoid the spread discussed in the previous section using a backoff for which its modulo T_s is 0; for example, after choosing a random backoff b, we subtract from it its value modulo T_s, i.e. $b - (b \bmod T_s)$.

We use the same scenario as for Figure 5.1, where the spread lead to an uncontrolled increase of receivers, with the following modifications:

The results are shown in Figure 5.5. Both the number of forwarders (67) and the number of receptions (2542) have decreased compared to the 1714 transimissions and 91 000 receptions from the original SLR protocol (Figure 5.4). A few collisions (93) still occur, without causing any problem.

Note that the latency incurred by the large backoffs should be negligible to usual applications: the timescale here is much smaller than traditional wireless networks.

To conclude, this method solves both propagation stops at the source and spreading (already proposed in sections 5.1.1.1, and 5.1.1.2); moreover, given that only a few copies are forwarded in each zone, it is extremely efficient at preserving network resources.

5.3.2/ SELF-CONFIGURABLE BACKOFF WINDOW BASED ON LOCAL DENSITY

The sleep mechanism improved with a counter-based forwarding described in the previous section uses a large backoff. The length of the backoff window, from which the backoff is chosen, is static, so someone needs to set it to a proper value. This value depends on the network density, so to make this configuration automatic, nodes need to know network density (number of their neighbours).

Therefore, we need an additional method to get the network density. The classical method relying on HELLO packets (where each receiver answers the initial packet) is prohibitively costly in communications in dense networks, and as such cannot be used in our case. To the best of our knowledge, the only algorithm efficient in dense networks (>100 neighbours) is DEDeN (Dense Estimator for Dense Networks) Arrabal et al. (2018a). This method is already presented in section 2.3.

The density estimation can then be used to set the backoff window used for the counter-based forwarding, similarly to Arrabal et al. (2018b)

$$w = 2(E + D)\rho, \tag{5.1}$$

where packet emission (sending) time $E = T_p(1 + \beta(s - 1))$ with s packet size in bits, propagation delay $D = CR/c$ with CR the communication range and c the speed of light (i.e. 3×10^9 m/s), and ρ is the density (number of neighbours). The backoff window is proportional to the density ρ and is chosen so that:

- The probability for neighbouring nodes to send a copy concurrently is very low (to give nodes sufficient time to decide whether other copies are sent or not).
- The average waiting time before a node among the neighbours starts to forward is also very low (to reduce delay). Indeed, even if the backoff window can be very large in dense areas, as the nodes are numerous and it is proportional to the density ρ, the probability that one of them chooses a small backoff is very high, so other nodes overhear its transmissions and cancel theirs. Thanks to this, the added delay and jitter are small, and, as a consequence, the memory needed to store packets is reduced too. A characterisation of the delay and the memory needed by nodes to buffer packets can be found in Arrabal et al. (2018b).

To evaluate the current method we use the same scenario as in the previous section 5.3.1,

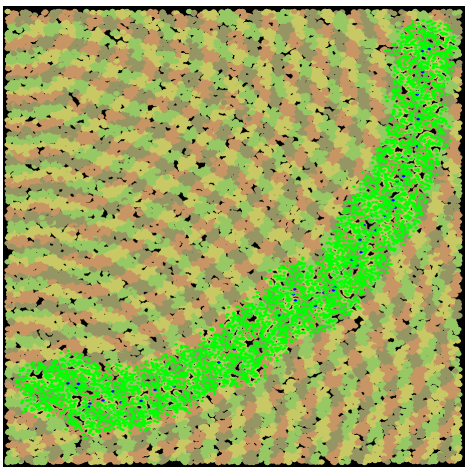


Figure 5.6: The receivers (green points) using backoff flooding Arrabal et al. (2018b) applied to SLR.

except using a dynamic backoff, as given by DEDeN. The number of transmissions (46) and the number of receptions (1761) are smaller than the method with a static window (67 and 2542 in the previous section), and the backoff window is also self-configurable.

Given that backoff flooding Arrabal et al. (2018b) is quite performant with respect to number of forwarders, we are interested in comparing it with our current method. We recall that, compared to backoff flooding, the method at this stage uses sleep and a restricted backoff, and is applied to SLR (instead of flooding). Whereas the number of forwarders (48) is, as expected, similar to our method (46), the number of receptions is significantly larger (13 033 vs 1761, as shown by the numerous green nodes in Figure 5.6 (vs Figure 5.5). Thus, backoff flooding alone, even if it highly reduces the number of forwarders, does not achieve our goal of reducing the number of receivers.

To conclude, the current method (sleep with dynamic counter-based forwarding) takes the best parts of the protocols it originates from: few senders (from counter-based approach) and few receptions (from sleep mechanism).

5.3.3/ SELF-CONFIGURABLE AWAKEN DURATION BASED ON LOCAL DENSITY

In the previous sections the current method was used only on homogeneous environments, with nodes uniformly randomly distributed in the area. In this section, we use a more realistic environment, heterogeneous, where the data flow crosses areas with widely different node densities. In such a network, nodes have different number of neighbours. If the duration of the awake interval (duty cycle) is the same for all the nodes (as it was the case until now), then in regions with small density the number of nodes awake simultaneously is small, which means that the propagation might stop, and in regions with high density the number of awake nodes is large, which means that numerous nodes

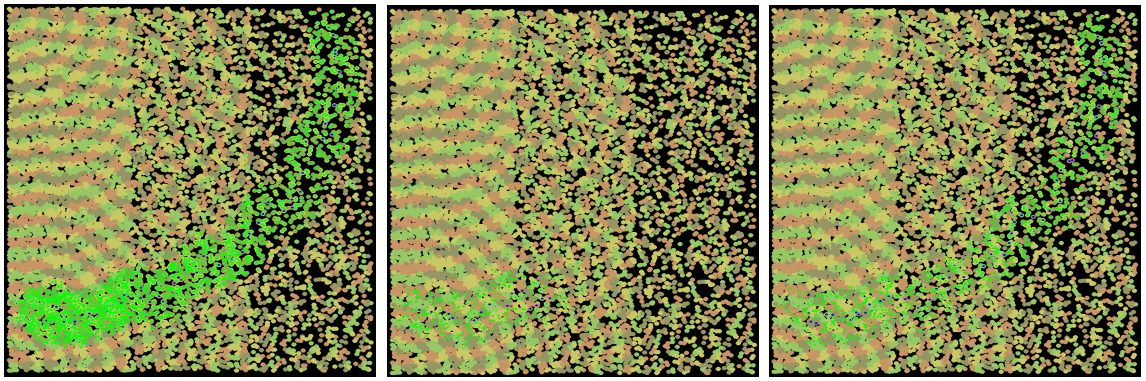


Figure 5.7: Receiving nodes (points in green) in counter-based forwarding: without sleep (left), with sleep and static awake duration (middle), and with sleep and a dynamic awake duration through a fixed number of awake nodes (right).

receive and process the packet.

The scenario is the same as previously section 5.3.1, except node placement, with three vertical sub-areas containing respectively 8000, 4000 and 2000 nodes (each of them uniformly distributed). The data flow (from bottom-left to top-right) crosses areas with different densities, but also the boundaries between them, where the density changes progressively. We recall that the scenario uses shadowing propagation and error model that affects the perceived neighbours and makes communications significantly more difficult.

To better understand the usefulness of the new method, we present progressively three counter-based methods.

The first one is counter-based SLR without sleep, hence all the nodes around the path receive the data packet. Figure 5.7 (left) clearly shows the effect of the three different density zones on the number of receivers (green points): the denser the area, the higher the number of receivers. In this example, there are 43 senders and 5780 receivers.

Next, the sleep mechanism is added, with a static awake duration, and the same for all nodes. As such, the number of neighbours awake at the same time is different among the nodes based on the density, e.g. nodes in less dense areas have 2 neighbours awake at the same time in average, whereas nodes in denser areas have 10 neighbours. As expected, a static value is not recommended for heterogeneous networks: if the value is too high, the packet reaches the destination at the cost of numerous receptions in the denser areas; if the value is too low, the packet stops propagating in the scarce areas. The later case is represented on Figure 5.7 (middle), where the data flow stops at the border between the high and medium density areas.

Finally, to avoid this heterogeneity in terms of number of neighbours awake concurrently, we do not use the same awake interval for all the nodes, but different values so that all

Table 5.3: Network scenario parameter

Number of nodes	15 000, randomly distributed
Awake duration	7000 fs (7% of T_s)
Data flow	100 unique packets crossing most of the network
Protocol	SLR

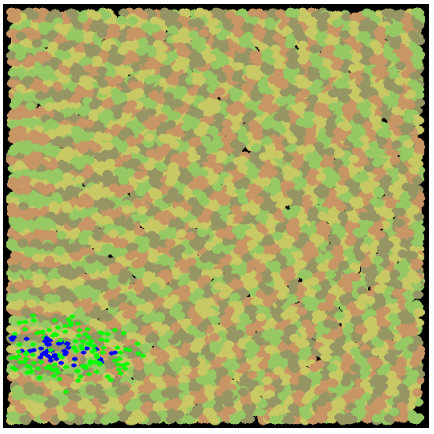


Figure 5.8: Propagation of a single packet stopping after a few hops.

nodes have a given number of awake neighbours, the same for all nodes. For example, a number of 5 awake neighbours in average might translate to an awake duration of 10% for nodes in less dense areas nodes, and of 2% for nodes in five times denser areas. Each node computes the awake duration aD (as a percentage of T_s) based on the given number of awake neighbours aN (e.g. 5) and its local density, using a simple formula a$D = aN/\rho$ (if $aD > 1$, then aD is set to 1), where the local density ρ can be obtained using a specific algorithm, such as DEDeN.

The number of awake neighbours aN must not be too small, for two reasons: not all the neighbours of a node are potential forwarders (but only those towards the destination) and the use of shadowing (section 5.1.2) rarefies the number of neighbours near the border. As a consequence, only few of the neighbouring nodes are found in the next SLR zone, where potential forwarders are. This can be seen in the following scenario 5.3.

The average number of neighbours is 300, so there are $7 \times 300/100 = 21$ nodes awake in average at any moment. This seems sufficient, however, due to shadowing and forwarding zone, only 10 to 20 neighbours are potential forwarders, so the number of awake potential forwarders is between $7 \times 10/100 = 0.7$ to 1.4 in average. Thus, there is a high risk of a packet being lost due to all possible forwarding nodes being asleep when it is transmitted. Figure 5.8 shows the propagation of one of the 100 data packets which quickly stops due to no node being awake on some part of the path.

Therefore, we need to use a sufficiently large aN, which is related to the number of neigh-

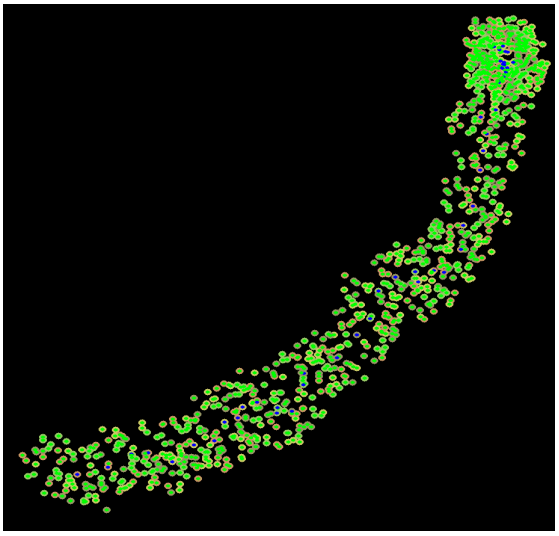


Figure 5.9: Smart repetition at destination zone.

bours, which is much greater than the number of potential forwarders. We use $aN = 50$. The results of this new method are shown in Figure 5.7 (right). The number of receptions is kept constant and low along the whole path, no matter the local density, and the packet successfully reaches the destination using only 44 senders and 2190 receivers.

It is worth to mention that in the right low-density sub-area, SLR zones contain only a few nodes (often 2 or 3), and still SLR plus counter-based forwarding and sleep algorithm makes packet reach the destination.

Compared to the 1714 senders and more than 91 000 receptions in the initial 30 000 node SLR-only scenario (section 5.1.4), in the given 14 000 node scenario the current method needs only 44 senders and 2190 receivers to route the packet from bottom-left to the top-right of the network. To sum up, the current method gives few receivers and is self-configurable, no matter the local density. It is thus a complete method to optimize routing phase from the number of receptions point of view.

5.3.4/ PROCESSING AT THE DESTINATION ZONE

By applying the previous improvements, the sleeping mechanism becomes able to reduce the number of packet forwarded and receptions. Those will positively affect in reducing nodes resources consumption, therefore lead to extend network lifetime.

Recall that by applying sleeping technique, when a packet arrives at the destination zone, the destination node may indeed be asleep and would miss the packet. A special work should be applied to increase the chance of packet receptions.

The retransmission algorithm already described in section 4.3.2.2 can be applied.

5.4/ CONCLUSION

We pinpointed a problem in dense multi-hop networks, which is a high number of receivers during routing in dense networks. We first showed that a nanonetwork-oriented fine-grained sleeping alone is not efficient in reducing the number of receptions in a several-hop path. Then, several challenges have been presented, such as propagation stopping early, efficiency dropping as the distance from the source increases (spreading), and adverse effects of heterogeneous environments. The root causes of these phenomena have been investigated and solutions have successively been proposed to all of them. The specific case of arrival to a destination node, which might sleep when the packet arrives in its zone, has been considered.

The final algorithm, consists of an improvement of SLR using sleeping with an adaptive awake duration, a counter-based forwarding with a discrete backoff, and a special mechanism at destination zone. The protocol has been evaluated in heterogeneous networks with tens of thousand of nodes with densities of up to a few hundreds neighbours, using a more realistic shadowing propagation model proves highly efficient. It reduces not only the number of transmissions to a close to optimal value, but also the overall number of receivers. The protocol is self-configurable, by reducing the number of receivers (and transmitters) of a flow to stable values no matter the local node density. As a consequence, this new protocol allows to increase the channel use in a dense network context without causing more losses.

6

REDUCING NETWORK CONGESTION BY SPLITTING TRAFFIC OVER MULTIPLE PATHS

Congestion control, whose aim is to make packets reach their destination efficiently, is being redefined in the context of networks of tiny nodes, both in detection and action to take. In multi-hop electromagnetic nanonetworks, congestion arises from the limited buffers of individual nodes on the path instead of limited channel bandwidth. Given the high network density, avoiding congestion can be done using a deviation-based routing technique. A previous attempt of deviation used one side to deviate and was successful in deviating packets. But deviating packets to one side only might lead to overloading the nodes along the path. In addition, if the packet collides in a second congested area, the packet will not deviate anymore and will not reach its intended destination. In the current work, we enhance the deviation by splitting traffic, randomly on the left and right sides, effectively making it use multiple paths and spreading the congestion over a larger area, hence minimizing it. Our evaluation shows the effectiveness of this deviation technique, allowing packets to reach their destination.

6.1/ INTRODUCTION

Recall that a nanonetwork can be a ultra-dense network, which we define as a network where nodes have high node density, they may have hundreds or thousands of neighbours in their communication range. With such a huge number of nodes distributed within the network, a new dilemma arises that will affect the effectiveness and the performance of the network, which is the congestion.

In a traditional dense sensor network, congestion is the state where the number of packets exchanged at some time exceeds the network capacity Jornet et al. (2012). Some of

them are discarded or lost due to collisions.

The are several differences between nanoscale and macroscale networks, such as the radio frequency used, the routing protocol, and the hardware capabilities. Focusing on nanonetworks, terahertz frequencies allow for higher bandwidth and throughput, up to several Tb/s Mohrehkesh and Weigle (2014).

In macroscale networks (internet), one of the well-known protocols that have a network congestion avoidance algorithm is TCP (Transmission Control Protocol).

The mechanisms are different when moving to nanoscale networks based on tiny nodes, equipped with embedded computing devices interfacing with sensors/actuators. They are used in indoor and outdoor applications to monitor a physical or environmental event. Contrary to traditional wireless networks, congestion in nanonetworks does not arise from a saturated channel, but from an insufficient capacity of individual nodes on the multi-hop path to process all the incoming concurrent packets Bicen and Akan (2011).

Nodes can monitor the number of concurrently received packets, which can be used as a parameter to indicate the level of congestion. Detecting a congestion at a node means that neighboring nodes are most probably experiencing the same conditions. This means that the whole area is congested, which leads to the aim of avoiding routing packets through it.

So nanonodes' limited resources lead to a decrease in the node's ability to process incoming packets. We can conclude that the congestion caused in nanonetworks is not related to the saturated channel phenomenon, but to the insufficient nodes' hardware capacity to process all the incoming packets.

To decrease congestion, the traditional solution has been either to decrease the rate of packets injected at the source, or to drop packets when nanonodes are unable to receive and store them due to buffer saturation. A novel solution to decrease congestion has recently been proposed, using a fine-grained sleeping mechanism Medlej et al. (2020).

An alternative solution is to deviate packets in case of congestion, given that network is sufficiently dense and provide several paths to reach the destination Arrabal et al. (2019). In case of congestion, routers use the field m in the packet, which specifies the distance of the path to follow compared to the direct path between source and destination. As such, a path of 1 represents the direct path, whereas a path of 3 represents the path at $3 - 1 = 2$ zones further than the direct path.

The drawback of this algorithm is that it makes packets deviate on only one side. By doing so, the nodes located along this side will be overloaded with time, and their hardware resources will be fast consumed. Also, the packets may encounter a second congested area on the deviated route.

The contribution of this work is to enhance the deviation algorithm proposed in Arrabal et al. (2019), and to evaluate its efficiency. This enhancement splits traffic randomly on both sides (left and right of the congested area) at each hop with congestion, giving multiple paths in a multi-hop network. The enhanced algorithm also solves the case of a second congested area during the packet routing. It also deviates the packet around the congested area, helping it to reach its intended destination. Note that the algorithm does not explicitly construct an end-to-end route, neither the primary nor the deviated one, which perfectly matches nanonetwork conditions, where nodes have very little memory and cannot store large data such as routing information.

The advantages presented by our approach to managing network congestion are: (1) It presents an original idea. (2) It compares the performance of both algorithms (original and enhanced), showing a noticeable minimization of congestion when using the enhanced algorithm. It's important to note that our work: (3) deals with the case of encountering a second congested area in the deviated path, which is a new idea that hasn't been suggested before. (4) Retransmissions are created on the node where the congestion occurs, hence the retransmitted packet does not uselessly travel from the source to the router again (as in the case where the retransmitted packet starts at the sender). (5) Last but not least, deviation happens as part of the routing algorithm. Therefore, neither the sender nor the receiver are concerned about message deviation during its transmission, which decreases the energy cost through low transmission delays even in the case of congestion.

6.2/ RELATED WORK

Congestion occurs when numerous packets have to be exchanged but nanonodes are not able to process them because of the limited resources. Congestion causes queuing delay, packet loss, or blocking of new connections, and leads to packet loss and interference, which affect network efficiency and lifetime Kurose (2005).

In the traditional network, congestion control and congestion avoidance represent a milestone in avoiding network collapse. They use various mechanisms such as exponential backoff in protocols such as CSMA/CA in 802.11 Savage et al. (1999), and CSMA/CD in the original Ethernet, window reduction in TCP, and fair queueing in devices such as routers and network switches.

The traditional solution is either to decrease the rate of packets injected at the source, or to drop packets when nanonodes are unable to receive and store them due to buffer saturation.

TCP includes various aspects of an additive increase / multiplicative decrease (AIMD)

scheme Kafi et al. (2014), along with other schemes such as slow start and congestion window. Over time, researchers have expanded the TCP protocol and produced several versions such as TCP New Reno, TCP Vegas, TCP Fast Parvez et al. (2010); Grieco and Mascolo (2004); Jin et al. (2004), focused on congestion avoidance techniques to solve the packet loss problem.

The approach is different when moving to nanoscale networks based on tiny nodes, equipped with embedded computing devices interfacing with sensors/actuators. Congestion in nanonetworks can have multiple causes, but the most common ones are either a low link bandwidth or the lack of network resources especially the buffer size in the nanonodes Jornet et al. (2012). Nanonetworks are used in indoor and outdoor applications to monitor a physical or environmental event. Depending on application, the upstream traffic delivery can be Kafi et al. (2014):

- Event-based: network load is light but becomes active in response to a detected event.
- Continuous sensing: some applications require continuous sending of sensing values, ex: nuclear monitoring.
- Query-driven: sink node invokes and queries sensing nodes to answer.
- Hybrid: bulk data is generated in addition to the constantly sensing data.

Congestion detection strategies are numerous Sergiou et al. (2014). The most used are packet loss (can be measured at the sender if ACK is used, also it can be measured at the receiver if sequence numbers are used), queue length (buffer usage in each node can serve as an indication of congestion, when the buffer exceeds a fixed threshold then the congestion is signaled), packet service time (service time is used to continuously adjust the rate at which children send their packets), the ratio between packet service and packet inter-arrival time (scheduler between network layer and MAC layer will quantify the number of packet scheduled per time unit, this ratio indicates node level and link level congestion, delay (quantifies the necessary time starting the packet generation at the sender until its successfully reception at the next hop receiver).

Recently, the interest in different types of networks, such as mobile ad hoc networks (MANET) and wireless sensor networks (WSN), lead to the proposal of new techniques of congestion control but with different concepts.

In Fang et al. (2010), authors note that congestion in wireless sensor networks (WSNs) does not only cause severe information loss, but also leads to excessive energy consumption. To address this problem, they proposed a novel scheme: congestion avoidance, detection and alleviation (CADA). As its name implies, the scheme contains relevant mechanisms for avoiding congestion proactively, detecting congestion timely and

alleviating congestion reactively. The scheme comprises three main mechanisms. Firstly, it attempts to suppress the source traffic from event area by carefully selecting a set of representative nodes to be data sources. Secondly, the onset of congestion is indicated in a timely way by jointly checking buffer occupancy and channel utilization. Lastly, the network attempts to alleviate congestion in the traffic hotspot by either resource control or traffic control, which is dependent on the specific congestion condition.

In Kazmi et al. (2022), to avoid the congestion, authors adjust the transmission rate at current node based on its traffic information (normalized queue length and congestion degree). Congestion degree and buffer occupancy ratio for different values of transmission rate are used to obtain the amount of retransmitted packets. Multiclassification is done to control the congestion using a data science technique, namely support vector machine (SVM). They have proposed two techniques, namely DE-SVM (Differential evolution) and GWO-SVM (Grey wolf optimization algorithm), to solve the problem of congestion and to classify the complex data.

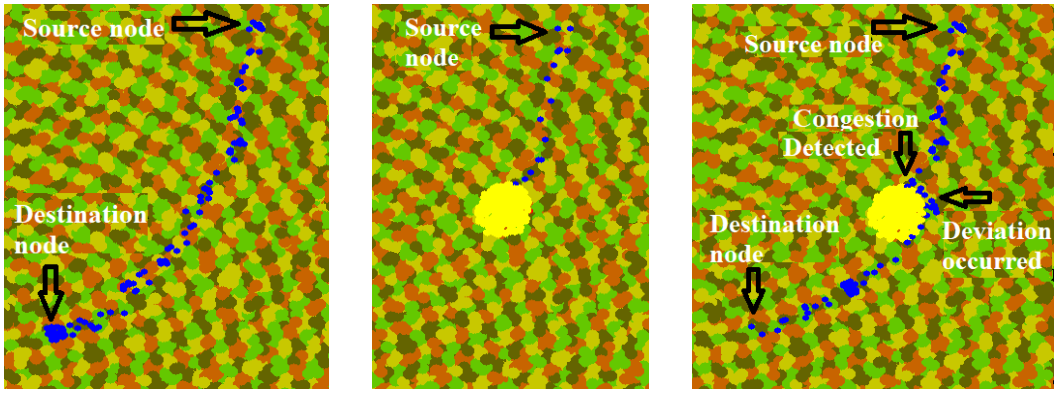
In many cases, a single parameter cannot accurately indicate congestion. The selection of such a parameter should be related to some factors such as network structure, application, and traffic nature, and used rate Kafi et al. (2014).

6.3/ ORIGINAL ONE-SIDE PACKET DEVIATION

In our laboratory, the nanonetwork team Arrabal et al. (2019) proposed a deviation-based congestion control. The main idea are: (1) to detect the congestion in nanonetworks, and (2) instead of decreasing the sending rate (to reduce congestion), to use a less saturated path to deliver the packet to the destination.

In this work, the routing protocol uses a variant of Stateless Linear Routing (SLR). SLR divides network area in SLR zones, in which nodes share the same coordinates Tsioliaridou et al. (2017), and data packets are routed in a linear routing path. In original SLR, all the nodes from zones on the path forward a packet. In dense zones, this leads to congestion for even one message initially sent. To avoid this situation, SLR is combined with backoff flooding, an efficient forwarding scheme designed to drastically reduce the number of transmitters while maintaining full coverage Arrabal et al. (2018b).

In the case of no congestion, SLR routes packets linearly from source SLR coordinate to destination SLR coordinate (Fig. 6.1(a)). Path-width value m is 1, so the path is one zone wide. Original SLR does not detect congestion, and hence propagation can stop (Fig. 6.1(b)). On the contrary, in deviation method Arrabal et al. (2019), when a node on the path detects increased congestion, it reacts by deviating the current packet by tweaking the path-width (increase m by 1, Fig. 6.1(c)). Therefore, the message ”spreads



(a) (b) (c)

Figure 6.1: Original deviation algorithm (messages propagate from top to the bottom): (a) no congestion, (b) congestion stops the message propagation, (c) deviation.

out" to occupy additional zones. This solution works in deviating the entire traffic to one side of the congested area.

6.4/ ENHANCED TWO-SIDE DEVIATION

The original deviation method Arrabal et al. (2019) deviates on only one side. When several packets deviate at the same side, congestion might occur on that side. Moreover, given that packets use one (deviated) path only, the congestion moves but stays the same. We need an algorithm which "spreads out" the congestion on a larger zone, hence reducing it and phase it out.

Our enhancement to the deviation is thus to split the traffic into several paths (multi-path deviation). This is done by changing the value of m from SLR and trying to force each packet of the flow to take a different route.

To the best of our knowledge, this idea has not been discussed yet. This enhanced algorithm will be executed when a node detects a congested area during the routing phase.

Recall that the congestion arises from the limited buffers of individual nodes on the path instead of limited channel bandwidth. A node can estimate its local congestion at a time t as follows:

$$c_n(t) = r_n(t)/MCR \tag{6.1}$$

where $r_n(t)$ represents the number of reception buffers of node n in use at time t, and MCR is the overall number of reception buffers of a node.

The congestion c_n needs to be compared to a congestion threshold c_U to know if a congestion has occurred. As soon as $c_n(t) > c_U$, nodes deviate the packets away from their

current path m of the SLR protocol. We want to spread out congestion, hence that copies of the same packet take the same path, but different packets take different paths. When nodes detect a congestion area, they split the traffic away from the congested area and spread it on several areas. This is done by changing the value of m, which forces packet to take a further route than the direct original one.

The enhanced deviation algorithm is presented in Algo. 3. The path-width m is the key in determining which path the packets will take. Firstly, the node tests if congestion is detected using Eq. 6.1. Next, the path-width m of SLR is computed using a hash function to make the routing decision. Three cases appear:

- Case 0: $m = 1$, the algorithm tries to decrement m, therefore deviating the packet to the left of the congested area.

- Case 1: $m = -1$, the algorithm tries to increment m, therefore deviating the packet to the right of the congested area.

- Default case: there is no congestion, in this case the packet resumes or goes back to its initial path.

After overcoming the congested area, m goes back to its initial value. This is done by incrementing m if it is lower than -1, and decrementing it if greater than 1.

Algorithm 3 Enhanced deviation algorithm.

INPUT:

MCR (maximum concurrent receptions)
c_U (congestion threshold)
c_L (congestion level)
A node checks if there is a congestion
$c_n(t) = r_n(t)/MCR$

if $c_n(t) > c_U$ **then**
switch on (hash)
case 0:
if m == 1 **then**
m = -1
else m = m - 1
endif
case 1:
if m == -1 **then**
m = 1
else m = m + 1
endif
default:
m gets back to its default value
end switch
elseif 100 * $c_n(t) < c_L \wedge ((m > 1) \vee (m < -1))$
if $m < -1$
m++
else m– –
endif
endif

hash is a hash function returning one value among three possible values (using modulo 3), but the same value for copies of the same packet. We aim that for a set of packets, half of them will deviate to the right of the congested area, while the other half to the left, and in case of no congestion the packet goes back to the direct path.

To summarize, compared to the various congestion control algorithms for wireless sensor networks (WSN) previously discussed in section 6.2, the majority of them (1) use a complex computation algorithm to sense the congestion occurrence as well as take the congestion control decision, (2) or use delayed transmission as a key in overcoming

Table 6.1: Simulation parameters.

Size of simulated area	6 mm * 6 mm
Number of nodes	20 000
Communication radius	350 µm
Average number of neighbours	203
β (TS-OOK time spreading ratio)	1000
T_p	100 fs
Packet size	100 bit
MCR (maximum concurrent receptions)	5
c_L/c_U	0.5 / 0.5

congestion, (3) or transmit the packet again from the sender. These methods might increase the node's energy consumption and lead to node performance degradation with time. Based on the listed reasons, the aim is to propose an algorithm that deals with congestion control, with minimum complexity and consequences on the node's resource consumption.

In our approach, neither sender nor receiver can detect that a message deviated during its transmission. This implies that deviating SLR cannot reduce traffic, or increase the transmission time delay as for example the case with TCP rate limiting. Instead, it performs a kind of load balancing, by randomly shifting the packet forwarding path (left and right) around the congested area, towards less utilized areas of the network. Doing so dispatches the congestion over a large area, making it negligible.

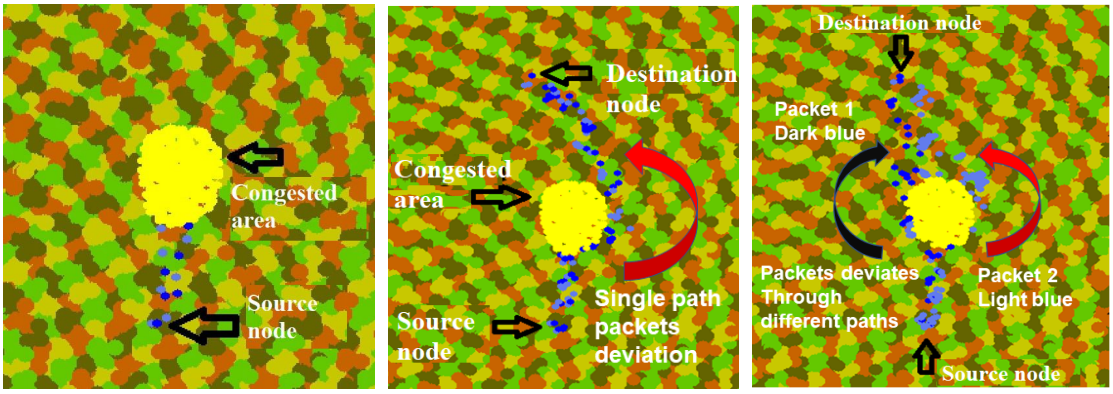
We recall that the algorithm does not explicitly construct any end-to-end route, neither primary nor deviated one. This feature matches nanonetwork constraints, where nodes have very little memory, contrarily to the discussed algorithms (section 2).

6.5/ EVALUATION

We evaluates the effectiveness of the enhanced deviation mechanism in bypassing network congestion. As a detailed analytic study is not possible, we instead evaluate the deviation mechanism through simulations.

The simulation parameters are shown in Table 6.1. The network is a 2D area, which allows two directions of deviation. The network has 20 000 nodes, which is considered dense. Nodes in the network can receive no more than 5 messages concurrently, i.e. MCR = 5.

To simulate a congested area, 17 concurrent flows are used. Each of them sends 13 packets, of size 1000 bits, with an interval time of 0 ns between each packet (i.e. sent one right after the other). These flows interfere with the message(s) from the source



(a) (b) (c)

Figure 6.2: The original vs enhanced algorithm for two packets: (a) congestion / no deviation, (b) deviation using the original algorithm, (c) deviation using the enhanced algorithm.

node to its intended destination, making them see a congestion.

The yellow circle in Fig. 6.2(a)-(c) represents the congested area. No additional messages can be received by the nodes in that area. Otherwise said, the node's buffers became saturated.

In our simulations we used the Stateless Linear Routing (SLR) 2.2 as a routing protocol.

Two packets are sent from bottom to top of the network.

We present results on the effectiveness of the enhanced algorithm, and afterwards a comparison with the original deviation algorithm.

We present results on the effectiveness of the enhanced algorithm, and a comparison with the original deviation algorithm.

Fig. 6.2(a) shows nodes forwarding using the original SLR protocol, that does not consider congestion detection and deviation. Therefore, the message cannot be forwarded through the congested area, nor deviate, and destination node misses the message.

Fig. 6.2(b) depicts the result while using the original deviation algorithm. It successfully detects the congestion. It also deviates the packets around the congested area. However, it can be seen that both packets deviate from one side, at the right of the congested area.

Fig. 6.2(c) shows the effectiveness of the enhanced deviation algorithm: nodes detect the congested area, deviate the packets around it from different sides (packet 1 from the left / dark blue dots, packet 2 from the right / light blue dots), route the packets successfully, and packets are correctly received by the destination node. Thus, it confirms that the deviation algorithm works as expected.

To compare the enhanced algorithm with the original one, we simulate another scenario, where the two packets cross two congested areas. Fig. 6.3 shows the result with the

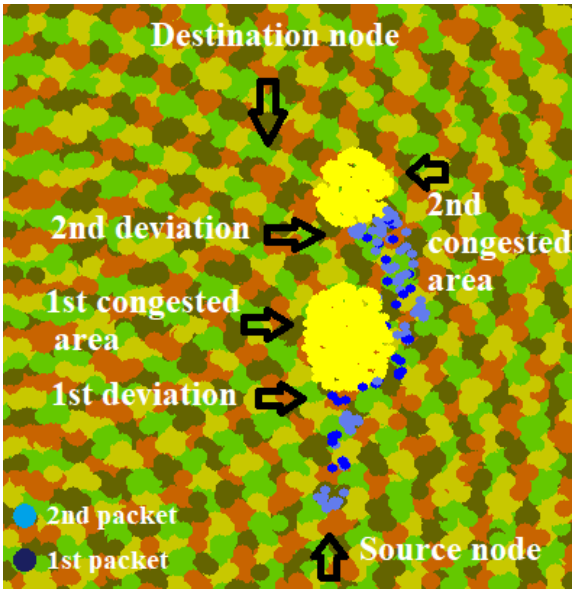


Figure 6.3: In original algorithm, deviation fails while observing a second congested area.

original deviation algorithm. Packets failed to deviate from the second congested area. The main reason for this problem is that sometimes the congested area is very large, then m needs to be increased by more than one hop to be able to complete its path to the destination.

Another point to be mentioned, in the initial algorithm, the location of the destination node is a key factor. As Fig. 6.3 shows, the destination node is located to the left of the second congested area. The deviation takes place to the right of the congested area. So, if we had the chance to pass the first congested area, the packet will not be able to overcome the second congested area.

Fig. 6.4 shows the result of the enhanced algorithm. The two packets deviate using both sides of the congested area. Packet 2 succeeds to overcome the second congested area; as the destination node is on the left side of the second congested area, the packet succeeds in reaching the destination zone, and received by its intended destination node. Recall that the location of the destination node is a key factor in packet reception. Comparing to the initial algorithm, deviating the packet to both sides of the congested area increases the chance of the packet reaching its destination in case of two congested areas. The algorithm shows high flexibility in dealing with two congestion areas.

Now, we compare both algorithms in terms of successfulness of at least one packet reception by the destination node as well as the average time elapsed between initial send and reception at the destination. To calculate the average time elapsed, each scenario is simulated 10 times with different seeds for the forwarding backoff. Table 6.2 shows the results of the evaluation. One side deviation and two sides deviation show different results while overcoming a congestion area.

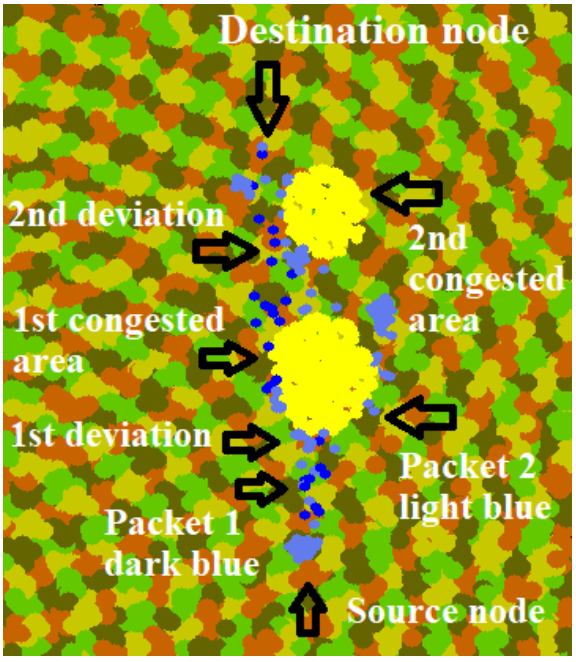


Figure 6.4: With enhanced algorithm, the deviation succeeds in overcoming the second congested area.

Starting with one congestion area, both algorithms succeed in delivering at least one packet to its intended node. But the elapsed time (1st received packet time – sent packet time, averaged for 10 runs) increases when using the two sides deviation (2.7 µs compared to 5.4 µs, and 2.9 µs compared to 3.6 µs). The cause is related to the random waiting times of SLR backoff, and to the different routing path taken by packets in the two algorithms. We consider that the increase in elapsed time is insignificant compared to the advantages offered by this algorithm.

The approach varies with the presence of two congestion areas. Table 6.2 shows the inefficiency of one side deviation in this case. The packet fails to reach its destination node (Fig. 6.3). On the other hand, due to the flexibility of the two sides deviation, at least one packet succeeds in reaching its destination (Fig. 6.4). The 1st packet is the one that had overcome the second congested area.

Last but not least, deviation happens as part of the routing algorithm. Therefore, neither the sender, nor the receiver are concerned about message deviation during its transmission, which decreases the energy cost through low transmission delays even in the case of congestion. Another advantage of our proposed algorithm is to increase the packet reception reliability which could be an important parameter in some applications. It also contributes to preserve the node's resources in terms of energy, by splitting the packet route into multi-paths, unlike the single path deviation process.

Table 6.2: Evaluation results for both deviation algorithms.

Algorithm	Successful reception	Elapsed time (average)
1 side dev/1 congestion area:		
1st packet	yes	2.7 µs
2nd packet	yes	2.9 µs
2 sides dev/1 congestion area:		
1st packet	yes	5.4 µs
2nd packet	yes	3.6 µs
1 side dev/2 congestion areas:		
1st packet	no	–
2nd packet	no	–
2 sides dev/2 congestion areas:		
1st packet	yes	7.6 µs
2nd packet	yes	5.1 µs

6.6/ CONCLUSION

We proposed and discussed an enhanced packet deviation algorithm within a nanonetwork. The goal is to overcome network congestion to ensure that packets reach destination. Our evaluation shows that the algorithm successfully delivers messages that otherwise are lost due to congestion. Deviation happens as part of regular forwarding, so messages deviate only if needed. Compared to the original deviation algorithm, our algorithm does not need new additions to the packet header. The value in what we have proposed lies in the ability of the packets to split its route around the congestion area, through multi-paths. By deviating packets through different paths, the algorithm contributes firstly in increasing the reliability of delivering the packets to their intended destination, and in distributing the load over several paths compared to the single path deviation algorithm. These features allow to reduce node's resources consumption and to increase network lifetime.

III

CONCLUSION AND PERSPECTIVE

This section comes as a general and comprehensive conclusion of what resulted from the research of three years in the field of nanonetworks that I conducted during my PhD study.

Firstly, I proposed a duty cycling technique embedded in the SLR routing protocol that aims to preserve the nodes' resources. We discussed the integration of the proposed *sleeping* mechanism into the SLR routing protocol and in the context of dense nanonetworks. Evaluations were conducted by using a detailed simulation tool. Evaluations illustrated a very peculiar behavior of wireless nanonetworks. By allowing an asynchronous sleep mechanism, we are able to improve significantly the amount of data that can be successfully transferred in the network in a given amount of time (we increase the usable capacity). The load is now dispatched because not all nodes participate in the local transmissions. Those will reflect positively on reducing resource consumption by the nodes (energy, CPU, memory ...).

The simulations I ran during the research clearly showed that the percentage of awake time needs tuning by considering the local network density and even the number of locally competing data flows. We have shown that an optimal value can be found, that optimizes the available capacity. The number of packets successfully transmitted increases while the number of ignored packets and collisions does not change much or even decreases. Each application/deployment can benefit very significantly from this tuning.

The network type varies with the application used. Therefore, it was necessary to study the feasibility of using the nano-sleep mechanism in heterogeneous networks (zones of different node densities). Because of the different density zones in the heterogeneous network, it is not efficient to set a fixed awaken duration for all nodes. Therefore, it is necessary to use an algorithm that helps to determine an automatic awaken duration of the node based on its neighbours density.

The idea is to embed a density estimator algorithm (DEDeN) to automatically tune the node awaken interval (awakenDuration). The estimator algorithm can be executed in different times (network deployment, or when a node needs to estimate its number of neighbours (which is represents our case).

Considerations have also been taken for the specific case of the destination zone (packet loss by the destination node because its sleep may coincide with the packet's arrival at the destination zone). An algorithm has been proposed in case the application needs data packets to reach a specific node in the destination zone. When the packet arrives to the destination zone, each receiving node retransmits the packet at the end of its awaken duration. In that case we contribute in increasing the chance of packet reception by the destination node in case it is sleeping.

The simulation proves the effectiveness of the sleep algorithm in heterogeneous net-

works. In the zone where the nodes' density is high, the node awaken duration will be low, while that duration will be high in the areas of low density. Also, the retransmission algorithm proves its efficiency in increasing the chance of the destination node to receive the packet when it is asleep while packet arrives at the destination zone.

A deep analysis of the destination zone shows a kind of congestion, when all the nodes at the destination zone will participate in the retransmission process. So a workaround should be proposed in order to solve this problem or at least to decrease its effect without loosing its benefit on packet reception.

I proposed a probabilistic retransmission algorithm at the destination zone where not all nodes participate in retransmission process. The goal is to ensure that the destination node receives the packet, while reducing the number of packets exchanged. In the proposed algorithm, the number of participating nodes is a function of the percentage of nodes awaken duration. Otherwise said, with a high percentage of awaken duration, the lowest the number of nodes participated in the retransmission process, and vice versa.

The evaluations show that probabilistically retransmiting the packet at the destination zone ensures high reliability in packet reception. Therefore, the benefit from this algorithm can significantly vary from one application to another. The simulation results show that the destination node is still able to receive the intended packet while decreasing the number of retransmissions at the destination zone, which contribute in reducing the congestion that might occur when all nodes will retransmit the packet.

After confirming the importance and impact of the sleep algorithm on preserving the nodes resources, by reducing the packet transmission rate and the number of participating nodes, thus contributing to an increase in the network lifetime, it was necessary to go into more depth and observe whether there was any improvement in the sleep mechanism.

The sleeping mechanism alone does not always reduce the number of receptions. We then proceed with an incremental approach to reveal, analyze, and fix one by one the routing inefficiencies incurred by this sleeping scheme. The final scheme mixes fine-grained duty-cycling with a self-configurable awake duration, a counter-based approach, a dynamic backoff with discrete values, and a particular algorithm at the destination zone (retransmission algorithm).

An important problem in dense multi-hop networks has been detected, which is a high number of receivers during routing in dense networks. I first showed that a nanonetwork-oriented fine-grained sleeping alone is not efficient in reducing the number of receptions. Then, several challenges have been presented, such as propagation stopping early, efficiency dropping as the distance from the source increase (spreading), and adverse effects of heterogeneous environments. The root causes of these phenomena have been

investigated and solutions have successively been proposed to all of them. The specific case of arrival to a destination node, which might sleep when the packet arrives in its zone, has been considered.

The final algorithm, an improvement of SLR using sleeping with an adaptive awake duration, a counter-based forwarding with a discrete backoff, and a special mechanism at destination zone, proves highly efficient. It reduces not only the number of transmissions to a close to optimal value, but also the overall number of receivers. The protocol is self-configurable, by reducing the number of receivers (and transmitters) of a flow to stable values no matter the local node density. As a consequence, this new protocol allows to increase the channel use in a dense network context without causing more losses. The protocol has been evaluated in heterogeneous networks with tens of thousand of nodes with densities of up to a few hundreds neighbours, using a more realistic shadowing propagation model.

Network congestion is considered one of the main performance degradation, were nodes and links are overloaded with traffic. In traditional networks, congestion cause is often related to latency, throughput, and bandwidth. This problem usually makes the end users' network slow. Contrary to traditional wireless networks, congestion in nanonetworks does not arise from a saturated channel, but from an insufficient capacity of individual nodes on the multi-hop path to process all the incoming concurrent packets.

To overcome congestion, the traditional solution has been either to decrease the rate of packets injected at the source or to drop packets when nanonodes are unable to receive and store them due to buffer saturation. We proposed an innovative solution which splits traffic randomly on both sides (left and right of the congested area) at each hop with congestion, giving multiple paths in a multi-hop network, with minimum complexity and consequences on the node's resource consumption.

Along the routing path, the sender or receivers cannot detect that a message deviated. In our approach, deviating SLR cannot reduce traffic, or increase the transmission time delay as for example the case with TCP rate limiting. Instead, the packet forwarding path is shifted around the congested area (left and right) to the less utilized and calm areas. By doing so, we dispatch the congestion over a large area, making it negligible.

Noting that the enhanced deviation algorithm does not explicitly construct any end-to-end route, neither primary nor deviated one. This feature matches nanonetwork constraints, where nodes have very little memory. Another positive point is that the algorithm proves its robustness with the possibility of having a second congested area. The evaluation shows that the packet succeeded in overcoming the congested area, and helped it complete its path to the intended destination.

However, it remains unclear whether the deviation solves some specific cases, but causes

even more trouble. For example, in a dense network scenario, with the use of sleeping mechanism, more evaluations are required to study the behavior of the deviation algorithm while a few nodes are awake.

The β value represents the time between two symbols. Symbols are sent during a period Ts and separated by a period T_p. T_p is much longer than T_s. The ratio T_s / T_p is called β and can be very large. The main case for a node timeout is when the packet cannot be forwarded by a further node. This may occur because of congestion. Sometimes, changing a network parameter could be a solution. Since the β value is considered one of the primary network parameters, then it's important to study the influence of changing the value of β on decreasing congestion and increasing the chance of successful packet reception.

Compared to the other nanonetwork simulator, BitSimulator software shows its effectiveness in allowing the simulation of high-density scenarios (hundred of thousands of nodes). Preserving nodes' resources and especially the energy, is considered a key point in extending the network lifetime. Hence the importance of improving the simulator with a parameter that calculates the level of remaining battery at each node. Thus, can make easy the way of evaluating the influence of an applied sleeping mechanism.

With the technological development that is happening now, it has become important to try to use and integrate technologies so that we can improve the performance or develop the idea presented. As a future work, I would like to ask about the ability to use artificial intelligence technology in increasing the ability of nanonodes computation especially in the results analysis field. This will open the door to improve the nodes' hardware performance and overcoming lacks.

Nodes' mobility is also considered an important key in any suggested system. Therefore, it is important to study the behavior of SLR routing protocol and the sleeping mechanism with nodes mobility. we should also give great importance to developing an energy harvesting technique that will increase the nodes' capabilities, and extend the network lifetime because the lack of energy could be the main problem in limiting a technology from its progress.

This thesis has laid the basis for future studies in nanonetworks while opening the door for mechanisms that aim to preserve node resource consumption (sleeping mechanism) and control network congestion by splitting packets to both sides of the congested area. Nanonetworks has a great impact in almost every field of our society ranging from health care to homeland security and environmental protection. This type of network is still very young, but *promising*.

BIBLIOGRAPHY

[Afaqui et al. 2017] AFAQUI, M. S. ; GARCIA-VILLEGAS, Eduard ; LOPEZ-AGUILERA, Elena: **"IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi"**. In *IEEE Wireless Communications* (2017), Jun, pages 130–137. DOI: 10.1109/MWC.2016.1600089WC

[Ali and Abu-Elkheir 2015] ALI, N. ; ABU-ELKHEIR, M.: **"Internet of nano-things healthcare applications: Requirements, opportunities, and challenges"**. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Los Alamitos, CA, USA : Networking and Communications (WiMob), Oct 2015, pages 9–14. DOI: 10.1109/WiMOB.2015.7347934

[Allhoff 2009] ALLHOFF, Fritz: **"On the Autonomy and Justification of Nanoethics"**. In *Nanotechnology & Society: Current and Emerging Ethical Issues*. Dordrecht : Springer Netherlands, Apr 2009, pages 3–38. DOI: 10.1007/978-1-4020-6209-4-1

[Arrabal et al. 2019] ARRABAL, Thierry ; BÜTHER, Florian ; DHOUTAUT, Dominique ; DEDU, Eugen: **"Congestion Control by Deviation Routing in Nanonetworks"**. In *6th ACM International Conference on Nanoscale Computing and Communication (NanoCom)*. Dublin, Ireland : ACM/IEEE, Sep 2019, pages 1–6

[Arrabal et al. 2018a] ARRABAL, Thierry ; DHOUTAUT, Dominique ; DEDU, Eugen: **"Efficient Density Estimation Algorithm for Ultra Dense Wireless Networks"**. In *27th International Conference on Computer Communications and Networks (ICCCN)*. Hangzhou, China : IEEE, Jul 2018, pages 1–9

[Arrabal et al. 2018b] ARRABAL, Thierry ; DHOUTAUT, Dominique ; DEDU, Eugen: **"Efficient Multi-hop Broadcasting in Dense Nanonetworks"**. In *17th IEEE International Symposium on Network Computing and Applications (NCA)*. Cambridge, MA, USA : IEEE, Nov 2018, pages 385–393

[Bicen and Akan 2011] BICEN, A O. ; AKAN, Ozgur B.: **"Reliability and Congestion Control in Cognitive Radio Sensor Networks"**. In *Ad Hoc Networks* (2011), number 7, pages 1154–1164. DOI: 10.1016/j.adhoc.2011.01.004

[Canovas-Carrasco et al. 2016] CANOVAS-CARRASCO, Sebastian ; GARCIA-SANCHEZ, Antonio-Javier ; GARCIA-SANCHEZ, Felipe ; GARCIA-HARO, Joan: **"Conceptual De-**

sign of a Nano-Networking Device". In *Sensors* (2016), Dec, pages 1–17. DOI: 10.3390/s16122104

[Chen et al. 2017] CHEN, Yawen ; WEN, Xiangming ; LU, Zhaoming ; SHAO, Hua ; JING, Wenpeng: **"Cooperation-Enabled Energy Efficient Base Station Management for Dense Small Cell Networks"**. In *Wirel. Netw.* (2017), Jul, pages 1611–1628. DOI: 10.1007/s11276-016-1234-y

[van Dam and Langendoen 2003] DAM, Tijs van ; LANGENDOEN, Koen: **"An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks"**. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. Los Angeles, California, USA : Association for Computing Machinery, Nov 2003, pages 171–180. DOI: 10.1145/958491.958512

[Dedu et al. 2014] DEDU, Eugen ; BOURGEOIS, Julien ; ZAINUDDIN, Muhammad A.: **"A First Study on Video Transmission Over a Nanowireless Network"**. In *Proceedings of ACM The First Annual International Conference on Nanoscale Computing and Communication*. Atlanta, GA, USA : Association for Computing Machinery, May 2014, pages 1–6. DOI: 10.1145/2619955.2619975

[Dhall 2017] DHALL, V.: **"An IoT Based Predictive Connected Car Maintenance Approach"**. In *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)* (2017), Mar, pages 16–22. DOI: 10.9781/ijimai.2017.433

[Dhoutaut et al. 2018] DHOUTAUT, Dominique ; ARRABAL, Thierry ; DEDU, Eugen: **"Bit-Simulator, an electromagnetic nanonetworks simulator"**. In *5th ACM/IEEE International Conference on Nanoscale Computing and Communication (NanoCom)*. Reykjavik, Iceland : ACM/IEEE, Sep 2018, pages 1–6

[Drexler et al. 1991] DREXLER, Eric ; PETERSON, Chris ; PERGAMIT, Gayle ; BRAND, Stewart: **"Unbounding the Future: The Nanotechnology Revolution"**. (1991)

[Fang et al. 2010] FANG, Wei-wei ; CHEN, Ji-ming ; SHU, Lei ; CHU, Tian-shu ; QIAN, De-pei: **"Congestion Avoidance, Detection and Alleviation in Wireless Sensor Networks"**. In *Journal of Zhejiang University SCIENCE C* (2010), Jan, number 1, pages 63–73. DOI: 10.1631/jzus.C0910204

[Feynman and Machinery 1960] FEYNMAN, RP ; MACHINERY, Infinitesimal: **"Herr et al."**. In *Engineering and Science* (1960), pages 22–36

[Ghadiry et al. 2012] GHADIRY, Mahdiar ; NADI, Mahdieh ; MOHAMMADI, Hosein ; BIN ABD MANAF, Asrulnizam: **"Analysis of a novel full adder designed for implementing in carbone nanotube technology"**. In *Journal of Circuits, Systems, and Computers* (2012), Aug. DOI: 10.1142/S0218126612500429

[Giraldo et al. 2019] GIRALDO, Juan P. ; WU, Honghong ; NEWKIRK, Gregory M. ; KRUSS, Sebastian: **"Nanobiotechnology approaches for engineering smart plant sensors"**. In *Nature nanotechnology* (2019), Jun, number 6, pages 541–553

[Grieco and Mascolo 2004] GRIECO, Luigi A. ; MASCOLO, Saverio: **"Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control"**. In *SIGCOMM Computer Commununication Revision* (2004), Apr, number 2, pages 25–38. DOI: 10.1145/997150.997155

[Hossain et al. 2018] HOSSAIN, Zahed ; XIA, Qing ; JORNET, Josep M.: **"TeraSim: An ns-3 Extension to Simulate Terahertz-band Communication Networks"**. In *Nano Communication Networks* (2018), Sep, pages 36–44. DOI: 10.1016/j.nancom.2018.08.001

[Jin et al. 2004] JIN, Cheng ; WEI, David X. ; LOW, Steven H.: **"FAST TCP: Motivation, Architecture, Algorithms, Performance"**. In *IEEE INFOCOM 2004*. Hong Kong, China : IEEE, Mar 2004, pages 2490–2501. DOI: 10.1109/INFCOM.2004.1354670

[Jornet and Akyildiz 2010] JORNET, Josep M. ; AKYILDIZ, Ian F.: **"Channel Capacity of Electromagnetic Nanonetworks in the Terahertz Band"**. In *2010 IEEE international conference on communications*. Cape Town, South Africa : IEEE, May 2010, pages 1–6. DOI: 10.1109/ICC.2010.5501885

[Jornet and Akyildiz 2011] JORNET, Josep M. ; AKYILDIZ, Ian F.: **"Low-weight Channel Coding for Interference Mitigation in Electromagnetic Nanonetworks in the Terahertz Band"**. In *2011 IEEE international conference on communications (ICC)*. Kyoto, Japan : IEEE, Jun 2011, pages 1–6. DOI: 10.1109/icc.2011.5962987

[Jornet and Akyildiz 2014] JORNET, Josep M. ; AKYILDIZ, Ian F.: **"Femtosecond-Long Pulse-Based Modulation for Terahertz Band Communication in Nanonetworks"**. In *IEEE Transactions on Communications* (2014), Apr, number 5, pages 1742–1754. DOI: 10.1109/TCOMM.2014.033014.130403

[Jornet et al. 2012] JORNET, Josep M. ; PUJOL, Joan C. ; PARETA, Josep S.: **"PHLAME: A Physical Layer Aware MAC Protocol for Electromagnetic Nanonetworks in the Terahertz Band"**. In *Nano Communication Networks* (2012), number 1, pages 74–81. DOI: 10.1016/j.nancom.2012.01.006

[Kafi et al. 2014] KAFI, Mohamed A. ; DJENOURI, Djamel ; BEN-OTHMAN, Jalel ; BADACHE, Nadjib: **"Congestion Control Protocols in Wireless Sensor Networks: A Survey"**. In *IEEE Communications Surveys & Tutorials* (2014), Mar, number 3, pages 1369–1390. DOI: 10.1109/SURV.2014.021714.00123

[Karn et al. 1990] KARN, Phil ; OTHERS: **"MACA-a new channel access method for packet radio"**. In *ARRL/CRRL Amateur radio 9th computer networking conference*, London, Canada, 1990, pages 134–140

[Kazmi et al. 2022] KAZMI, Hafiza Syeda Z. ; JAVAID, Nadeem ; AWAIS, Muhammad ; TAHIR, Muhammad ; SHIM, Seong-o ; ZIKRIA, Yousaf B.: **"Congestion Avoidance and Fault Detection in WSNs Using Data Science Techniques"**. In *Transactions on Emerging Telecommunications Technologies* (2022), Mar, number 3, pages e3756. DOI: 10.1002/ett.3756

[Kim et al. 2010] KIM, Betty Y. ; RUTKA, James T. ; CHAN, Warren C.: **"Nanomedicine"**. In *New England Journal of Medicine* (2010), number 25, pages 2434–2443

[Kurose 2005] KUROSE, James F.: *Computer Networking: A Top-Down Approach Featuring the Internet, 3/E*. Pearson Education India, 2005

[Lee et al. 2010] LEE, W.S. ; ALCHANATIS, V. ; YANG, C. ; HIRAFUJI, M. ; MOSHOU, D. ; LI, C.: **"Sensing Technologies for Precision Specialty Crop Production"**. In *Computers and electronics in agriculture*. Elsevier, Oct 2010, pages 2–33. DOI: 10.1016/j.compag.2010.08.005

[Li et al. 2018] LI, Jun ; WANG, Hao ; WANG, Xiumin ; LI, Zhengquan: **"Optimized sleep strategy based on clustering in dense heterogeneous networks"**. In *EURASIP Journal on Wireless Communications and Networking* (2018), Dec, pages 1–10. DOI: 10.1186/s13638-018-1311-2

[Maksimović 2017] MAKSIMOVIĆ, Mirjana: **"The Roles of Nanotechnology and Internet of Nano Things in Healthcare Transformation"**. In *in TecnoLógicas* (2017), Sep, pages 139–153. DOI: 10.22430/22565337.720

[Medlej et al. 2020] MEDLEJ, Ali ; BEYDOUN, Kamal ; DEDU, Eugen ; DHOUTAUT, Dominique: **"Scaling up Routing in Nanonetworks with Asynchronous Node Sleeping"**. In *28th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. Hvar, Croatia : IEEE, Sep 2020, pages 177–182

[Miller et al. 2005] MILLER, M. J. ; SENGUL, C. ; GUPTA, I.: **"Exploring the Energy-Latency Trade-off for Broadcasts in Energy-saving Sensor Networks"**. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS)*. Columbus, OH, USA : IEEE, Jun 2005, pages 17–26. DOI: 10.1109/ICDCS.2005.35

[Mohrehkesh and Weigle 2014] MOHREHKESH, Shahram ; WEIGLE, Michele C.: **"Optimizing Energy Consumption in Terahertz Band Nanonetworks"**. In *IEEE Journal on Selected Areas in Communications* (2014), Dec, number 12, pages 2432–2441. DOI: 10.1109/JSAC.2014.2367668

[Monteleoni et al. 2004] MONTELEONI, Claire ; BALAKRISHNAN, Hari ; FEAMSTER, Nick ; JAAKKOLA, Tommi: **"Managing the 802.11 energy/performance tradeoff with machine learning"**. (2004)

[Nan et al. 2012] NAN, Guofang ; SHI, Guanxiong ; MAO, Zhifei ; LI, Minqiang: **"CDSWS: coverage-guaranteed distributed sleep/wake scheduling for wireless sensor networks"**. In *EURASIP Journal on Wireless Communications and Networking* (2012), Feb, pages 1–14. DOI: 10.1186/1687-1499-2012-44

[Nichols and Bae 2012] NICHOLS, Joseph W. ; BAE, You H.: **"Odyssey of a cancer nanoparticle: From injection site to site of action"**. In *Nano today* (2012), Dec, number 6, pages 606—618. DOI: 10.1016/j.nantod.2012.10.010

[Parvez et al. 2010] PARVEZ, N. ; MAHANTI, A. ; WILLIAMSON, C.: **"An Analytic Throughput Model for TCP NewReno"**. In *IEEE/ACM Transactions on Networking* (2010), Mar, number 2, pages 448–461. DOI: 10.1109/TNET.2009.2030889

[Piro et al. 2013] PIRO, Giuseppe ; GRIECO, Luigi A. ; BOGGIA, Gennaro ; CAMARDA, Pietro: **"Nano-Sim: Simulating Electromagnetic-Based Nanonetworks in the Network Simulator 3"**. In *6th International ICST Conference on Simulation Tools and Techniques (SimuTools)*. Cannes, France : ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Mar 2013, pages 203–210

[Pyles et al. 2012] PYLES, Andrew J. ; QI, Xin ; ZHOU, Gang ; KEALLY, Matthew ; LIU, Xue: **"SAPSM: Smart Adaptive 802.11 PSM for Smartphones"**. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. New York, NY, USA : Association for Computing Machinery, Sep 2012, pages 11–20. DOI: 10.1145/2370216.2370219

[Reina et al. 2015] REINA, D.G. ; TORAL, S.L. ; JOHNSON, P. ; BARRERO, F.: **"A survey on probabilistic broadcast schemes for wireless ad hoc networks"**. Elsevier, Feb 2015, pages 263–292. DOI: 10.1016/j.adhoc.2014.10.001

[Saha et al. 2015] SAHA, Swetank K. ; DESHPANDE, Pratik ; INAMDAR, Pranav P. ; SHESHADRI, Ramanujan K. ; KOUTSONIKOLAS, Dimitrios: **"Power-throughput tradeoffs of 802.11n/ac in smartphones"**. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. Hong Kong, China : IEEE, Apr 2015, pages 100–108. DOI: 10.1109/INFOCOM.2015.7218372

[Savage et al. 1999] SAVAGE, Stefan ; CARDWELL, Neal ; WETHERALL, David ; ANDERSON, Tom: **"TCP Congestion Control with a Misbehaving Receiver"**. (1999), Oct, number 5, pages 71–78. DOI: 10.1145/505696.505704

[Sergiou et al. 2014] SERGIOU, Charalambos ; ANTONIOU, Pavlos ; VASSILIOU, Vasos: **"A Comprehensive Survey of Congestion Control Protocols in Wireless Sensor**

Networks". In *IEEE Communications Surveys & Tutorials* (2014), Apr, number 4, pages 1839–1859. DOI: 10.1109/COMST.2014.2320071

[Seyedi et al. 2013] SEYEDI, MirHojjat ; KIBRET, Behailu ; LAI, Daniel T. H. ; FAULKNER, Micheal: **"A Survey on Intrabody Communications for Body Area Network Applications"**. In *IEEE Transactions on Biomedical Engineering* (2013), Aug, number 8. DOI: 10.1109/TBME.2013.2254714

[Singh and Raghavendra 1998] SINGH, Suresh ; RAGHAVENDRA, Cauligi S.: **"PAMAS—power aware multi-access protocol with signalling for ad hoc networks"**. In *ACM SIGCOMM Computer Communication Review* (1998), Jul, number 3, pages 5–26. DOI: 10.1145/293927.293928

[Stelzner et al. 2016] STELZNER, Marc ; LAU, Florian-Lennert ; FREUNDT, Katja ; BÜTHER, Florian ; NGUYEN, Mai L. ; STAMME, Cordula ; EBERS, Sebastian: **"Precise Detection and Treatment of Human Diseases Based on Nano Networking"**. In *11th EAI International Conference on Body Area Networks (BodyNets)*. Turin, Italy : ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Dec 2016, pages 58–64

[Taniguchi 1974] TANIGUCHI, Norio: **"On the Basic Concept of Nanotechnology"**. In *Proceeding of the ICPE* (1974)

[Thalamy et al. 2019] THALAMY, Pierre ; PIRANDA, Benoît ; BOURGEOIS, Julien: **"A survey of autonomous self-reconfiguration methods for robot-based programmable matter"**. In *Robotics and Autonomous Systems* (2019), Oct, pages 103242. DOI: 10.1016/j.robot.2019.07.012

[Timmons and Scanlon 2011] TIMMONS, N.F. ; SCANLON, W.G.: **"Improving the ultra-low-power performance of IEEE 802.15.6 by adaptive synchronisation"**. In *IET Wireless sensor systems* (2011), Sep, pages 161–170. DOI: 10.1049/iet-wss.2011.0036

[Toffoli and Margolus 1991] TOFFOLI, Tommaso ; MARGOLUS, Norman: **"Programmable matter: Concepts and realization"**. In *Physica D Nonlinear Phenomena* (1991), pages 263–272

[Tsioliaridou et al. 2017] TSIOLIARIDOU, Ageliki ; LIASKOS, Christos ; DEDU, Eugen ; IOANNIDIS, Sotiris: **"Packet Routing in 3D Nanonetworks: A Lightweight, Linear-Path Scheme"**. Elsevier, 2017, pages 63–71. DOI: 10.1016/j.nancom.2017.01.001

[Wegner et al. 2006] WEGNER, Theodore H. ; WINANDY, Jerrold E. ; RITTER, Michael A.: **"Nanotechnology opportunities in residential and non-residential construction"**. In *NICOM 2: 2nd International Symposium on Nanotechnology in Construction*, RILEM, 2006, pages 339–347

[Yao et al. 2019] YAO, Xin-Wei ; HUANG, Wei ; OTHERS: **"Routing Techniques in Wireless Nanonetworks: A Survey"**. In *Nano Communication Networks* (2019), Sep, pages 100–250. DOI: 10.1016/j.nancom.2019.100250

[Ye et al. 2002] YE, Wei ; HEIDEMANN, John ; ESTRIN, Deborah: **"An Energy-Efficient MAC Protocol for Wireless Sensor Networks"**. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies*. New York, NY, USA : IEEE, Jun 2002, pages 1567–1576. DOI: 10.1109/INFCOM.2002.1019408

[Yick et al. 2008] YICK, Jennifer ; MUKHERJEE, Biswanath ; GHOSAL, Dipak: **"Wireless sensor network survey"**. In *Computer Networks* (2008), Aug, pages 2292–2330. DOI: 10.1016/j.comnet.2008.04.002

[Yin et al. 2008] YIN, G. ; YANG, G. ; YANG, W. ; ZHANG, B. ; JIN, W.: **"An Energy-Efficient Routing Algorithm for Wireless Sensor Networks"**. In *2008 International Conference on Internet Computing in Science and Engineering (ICICSE)*. Harbin, China : IEEE, Jan 2008, pages 181–186. DOI: 10.1109/ICICSE.2008.76

[Zhao and Govindan 2003] ZHAO, Jerry ; GOVINDAN, Ramesh: **"Understanding Packet Delivery Performance in Dense Wireless Sensor Networks"**. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. Los Angeles, California, USA : Association for Computing Machinery, Nov 2003, pages 1–13. DOI: 10.1145/958491.958493

[Zheng et al. 2011] ZHENG, Wei ; SHIH, Hui-Ru ; LOZANO, Karen ; MO, Yi-Lung: **"Impact of Nanotechnology on Future Civil Engineering Practice and its Reflection in Current Civil Engineering Education"**. In *Journal of Professional Issues in Engineering Education and Practice* (2011), Jul, pages 162–173

List of Figures

List of Tables

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — https://github.com/gallandarakhneorg/tex-templates
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Efficient algorithms for improving packet routing and congestion control in dense electromagnetic nanonetworks
Algorithmes efficaces pour améliorer le routage des paquets et le contrôle de la congestion dans les nanoréseaux électromagnétiques denses

Keywords: Nanonetwork, Congestion, Deviation, Packet Retransmission

Abstract:

Research has long focused on the possibility of miniaturizing machines and equipment while maintaining their operational capacity as much as possible. Nanoscale machines are considered a model of the development that has taken place at the level of miniaturizing machines. These are characterized by their limited resources (CPU, memory, battery) and computational capabilities. Nano-scale communication is characterized by its high bandwidth (Tb/s).
The work carried out during this thesis allowed the development of algorithms adapted to the nanoscale environment. An asynchronous duty cycling technique to preserve nodes' resources consumption and help in decrease congestion (in a homogeneous network); a dynamic awaken duration technique for nodes (based on a density estimation algorithm in a heterogeneous network), accompanied with a packet retransmission algorithm at the destination zone; an enhanced algorithm for packet retransmission based on precalculated probability; a packet splitting algorithm to overcome congestion area, and increase packet reception; finally, an improvement of a sleeping method to decrease the number of forwarders.
A detailed analytical study is not possible because nanomachines have not been built yet. So a real evaluation is not possible. For that reason, we are going to simulate our ideas using a simulator (BitSimulator) already developed by the nano-group at Franche-Comté university.

Titre : Algorithmes efficaces pour améliorer le routage des paquets et le contrôle de la congestion dans les nanoréseaux électromagnétiques denses

Mots-clés : Nanoréseau, congestion, déviation, retransmission de paquets

Résumé :

La recherche s'est longtemps concentrée sur la possibilité de miniaturiser les machines et équipements tout en conservant au maximum leur capacité opérationnelle. Les machines à l'échelle nanométrique sont considérées comme un modèle du développement qui a eu lieu au niveau des machines de miniaturisation. Ceux-ci se caractérisent par leurs ressources limitées (CPU, mémoire, batterie) et leurs capacités de calcul. La communication à l'échelle nanométrique se caractérise par sa bande passante élevée (Tb/s).
Les travaux menés au cours de cette thèse ont permis le développement d'algorithmes adaptés à l'environnement nanométrique. Une technique de cycle de service asynchrone pour préserver la consommation des ressources des nœuds et aider à réduire la congestion (dans un réseau homogène); une technique de durée d'éveil dynamique des nœuds (basée sur un algorithme d'estimation de densité dans un réseau hétérogène), accompagnée d'un algorithme de retransmission de paquets au niveau de la zone de destination ; un algorithme amélioré pour la retransmission de paquets basé sur une probabilité précalculée ; un algorithme de fractionnement de paquets pour surmonter la zone de congestion et augmenter la réception de paquets ; enfin, une amélioration d'une méthode de sommeil pour diminuer le nombre de retransmetteurs.
Une étude analytique détaillée n'est pas possible car les nanomachines n'ont pas encore été construites. Une véritable évaluation n'est donc pas possible. C'est pourquoi nous allons simuler nos idées à l'aide d'un simulateur (BitSimulator) déjà développé par le groupe nano de l'université de Franche-Comté.

Université Bourgogne-Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France